Altera PHYLite for Parallel Interfaces IP Core User Guide

2015.06.12

ug_altera_phylite

Subscribe Send Feedback

The Altera PHYLite for Parallel Interfaces IP core controls the strobe-based capture I/O elements in Arria® 10 devices. Use each instance of the IP core to support an interface with up to 18 individual data/strobe capture groups. Each group can contain up to 48 data I/Os as well as the strobe capture logic.

Device Family Support

The Altera PHYLite for Parallel Interfaces IP core supports Arria® 10 devices only.

For Arria V, Cyclone® V, and Stratix® V devices, use the ALTDQ_DQS2 IP core instead.

Related Information

- **ALTDQ_DQS2 IP Core User Guide**
  For more information about the ALTDQ_DQS2 IP core

Features

The Altera PHYLite for Parallel Interfaces IP core:

- Supports input, output, and bidirectional data channels
- Supports DQS-group based data capture, with up to 48 I/Os (including strobes) per group and DQS gating/ungating circuitry for strobe-based interfaces
- Supports output delays via interpolator
- Supports dynamic on-chip termination (OCT) control
- Supports quarter-rate to half-rate and half-rate to full-rate conversions. Also supports input, output, and read/DQS/OCT enable paths
- Supports single data rate (SDR) and double data rate (DDR) at the I/Os
- Supports PHY clock tree
- Supports dynamically reconfigurable delay chains using Avalon interface
- Supports process, voltage, and temperature (PVT) or non-PVT compensated input and DQS delay chains

Note: The non-PVT compensated component of the input delay is not set in the Quartus II software version 14.1 and will only be set in a future release of the Quartus II software.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

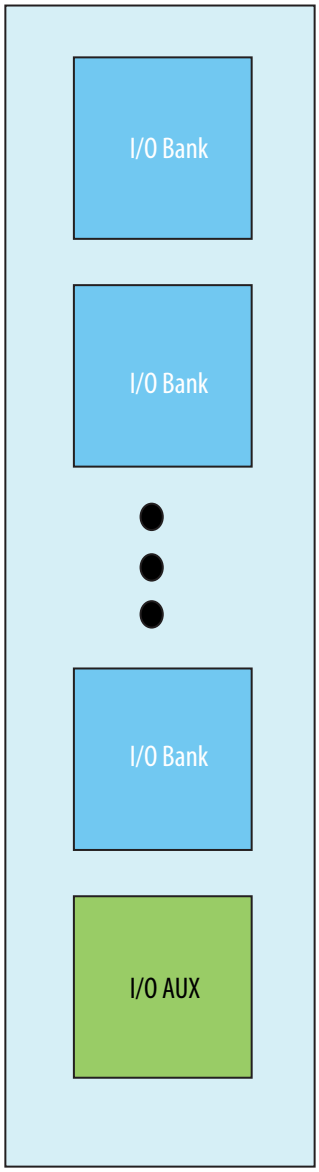
ISO 9001:2008 Registered

101 Innovation Drive, San Jose, CA 95134

Overview

The Arria 10 I/O subsystem is located in the I/O columns. Each column consists of up to 13 I/O banks and one I/O aux.

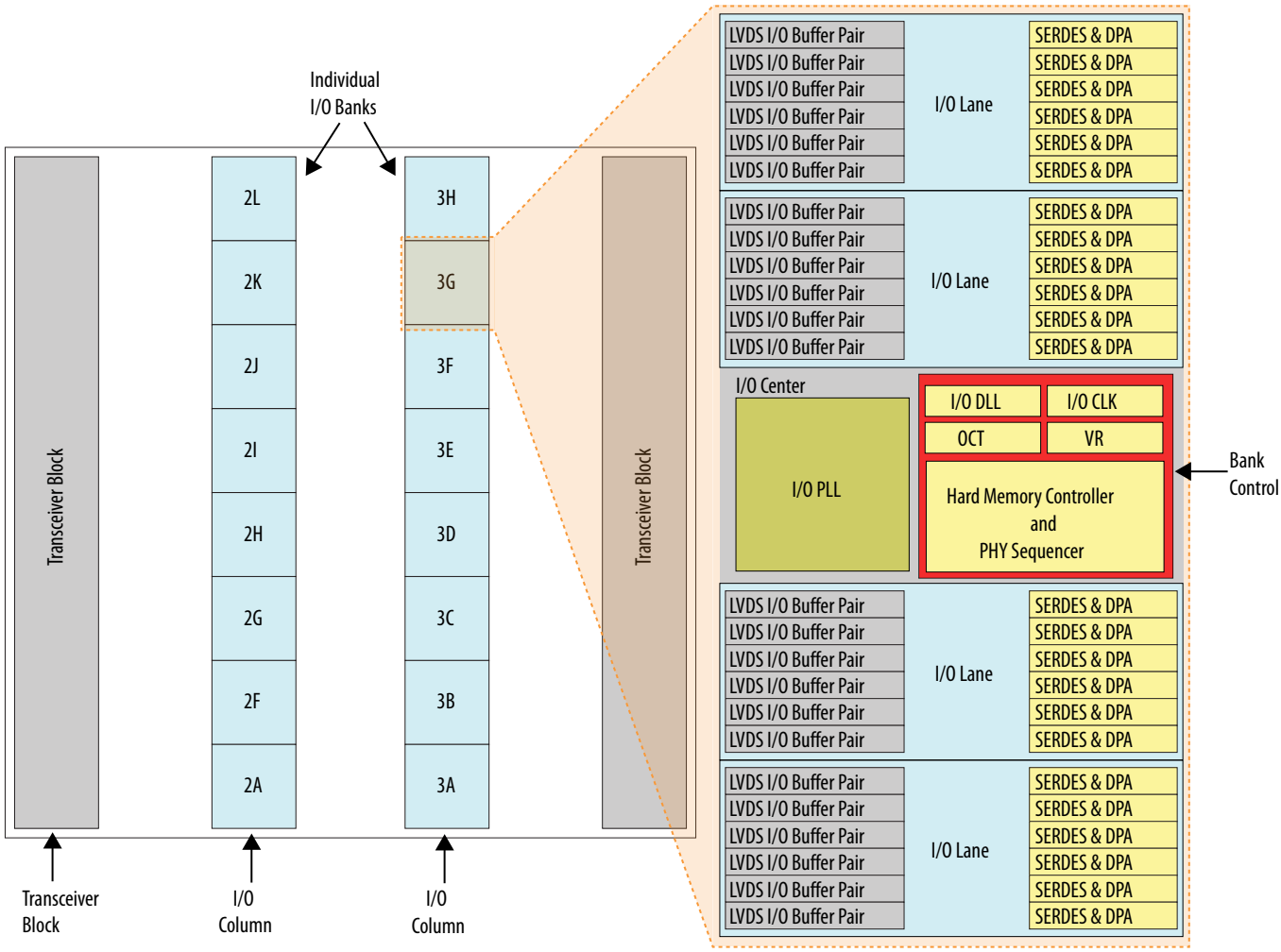
Figure 1: I/O Column for Arria 10 Devices



Each bank is a group of 48 I/O pins, organized into four I/O lanes with 12 pins for each lane.

Figure 2: I/O Banks in Arria 10 Devices

This figure shows a detailed view of the I/O bank in Arria 10 devices.



Each I/O lane contains the DDR-PHY input and output path logic for 12 I/Os as well as a DQS logic block. All four lanes in a bank can be combined to form a single data/strobe group or up to four groups in the same interface. Under certain conditions, two groups from different interfaces can also be supported in the same bank.

Related Information

- **Placement Restrictions** on page 15
  For more information about placement restrictions
- **Functional Description—Arria 10 EMIF**
  For more information about the architecture

Clocks

The Altera PHYLite for Parallel Interfaces IP core uses four clock domains for the output and input paths.

Table 1: Altera PHYLite IP Core Clock Domains

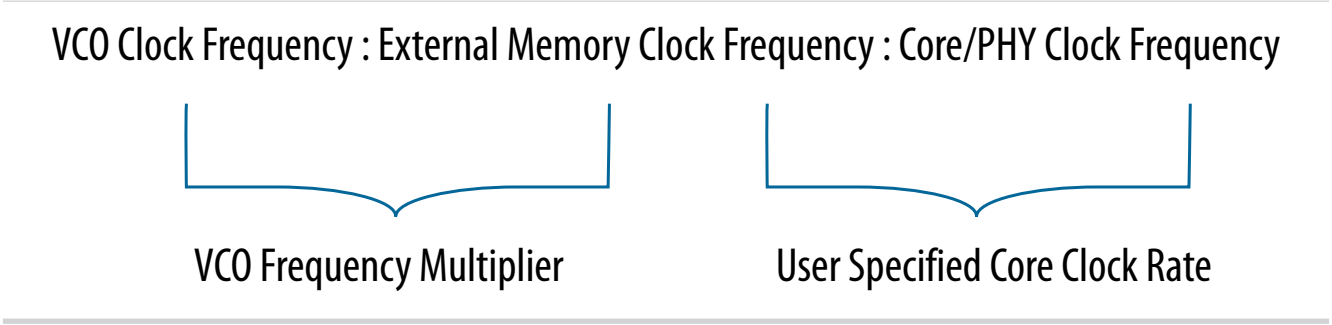
Clock Domain	Description
Core clock	This clock is generated internally by the IP core and it is used for all transfers between the FPGA core fabric and I/O banks.
PHY clock	This clock is used internally by the IP core for PHY circuitry running at the same frequency as the core clock. The PHY circuitry ensures that this clock is kept in phase with the core clock for core-to-periphery and periphery-to-core transfers.
VCO clock	This clock is generated internally by the PLL. It is used by both the input and output paths to generate PVT compensated delays.
External Memory clock	This is the user specified frequency at which the FPGA I/Os connected to the external device operate.

Related Information

Interface on page 5

Clock Frequency Relationships

Figure 3: Clock Frequency Relationships


VCO Clock Frequency : External Memory Clock Frequency : Core/PHY Clock Frequency
VCO Frequency Multiplier
User Specified Core Clock Rate

VCO Frequency Multiplication Factor

Table 2: VCO Frequency Multiplication Factor

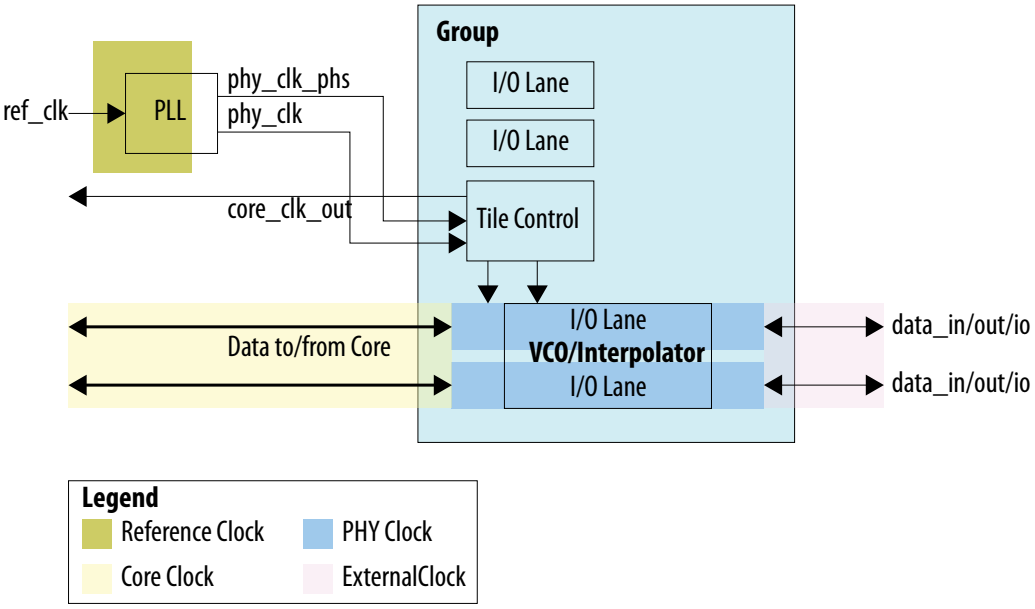
The relationship between the VCO clock frequency and the user specified external memory clock frequency is calculated during generation of the IP core based on the this table.

VCO Frequency Multiplication Factor	Speed Grade -1		Speed Grade -2		Speed Grade -3	
	Minimum Frequency	Maximum Frequency	Minimum Frequency	Maximum Frequency	Minimum Frequency	Maximum Frequency
1	600	1333.33	600	1333.33	600	1250
2	300	600	300	600	300	600
4	150	300	150	300	150	300
8	100	150	100	150	100	150

Interface

Figure 4: Top-Level Interface

This figure shows the top-level diagram of the Altera PHYLite for Parallel Interfaces IP core interface.



The Altera PHYLite for Parallel Interfaces IP core consists of the following interfaces:

- Clocks and reset
- Core data and control (broken down into input and output paths)
- I/O (broken down into input and output paths)
- Avalon configuration bus

Related Information

- **Output Path** on page 5
  For more information about the output path
- **Input Path** on page 8
  For more information about the input path
- **Signals** on page 38
  For more information about core data, control, and I/O interfaces signals

Output Path

The output path consists of a FIFO and an interpolator.

Figure 5: Output Path

This figure shows the output path for the Altera PHYLite for Parallel Interfaces IP core.

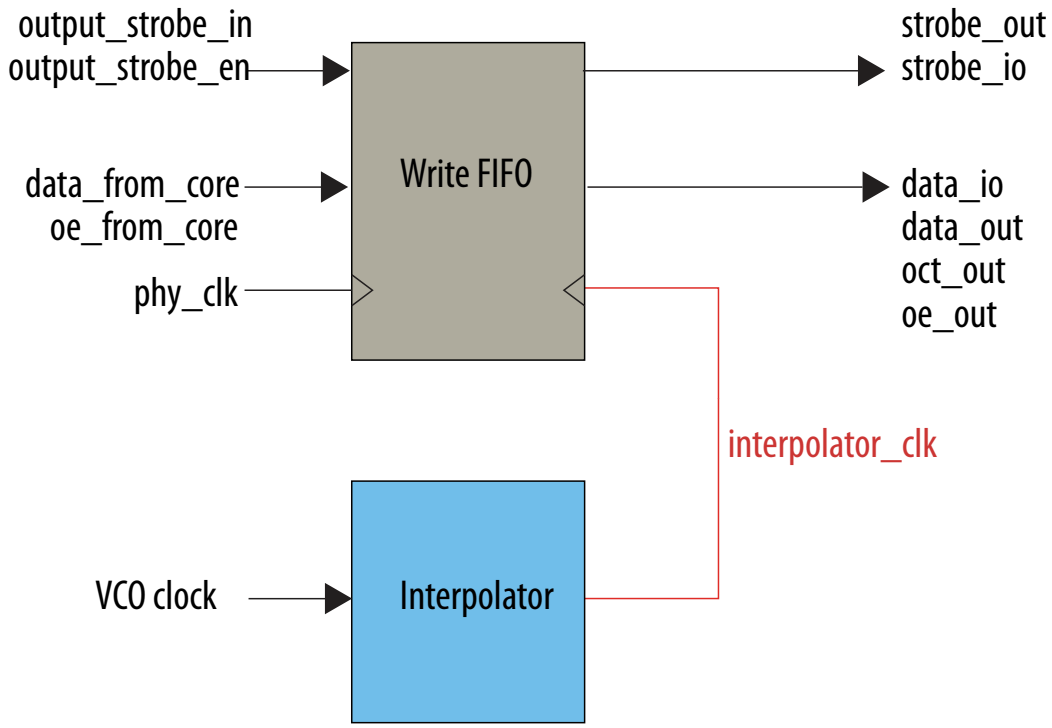


Table 3: Blocks in Output Path

This table lists the blocks in the output path.

Block	Description
FIFO	Serializes the output data from the core with a serialization factor of up to 8 (in DDR quarter-rate).
Interpolator	Works with the FIFO block to generate the desired output delay. You can dynamically configure the delay through the Avalon interface. For more information, refer to Dynamic Reconfiguration section.

The following figures show the waveform diagrams for the output path.

Figure 6: Output Path — Write Latency 0

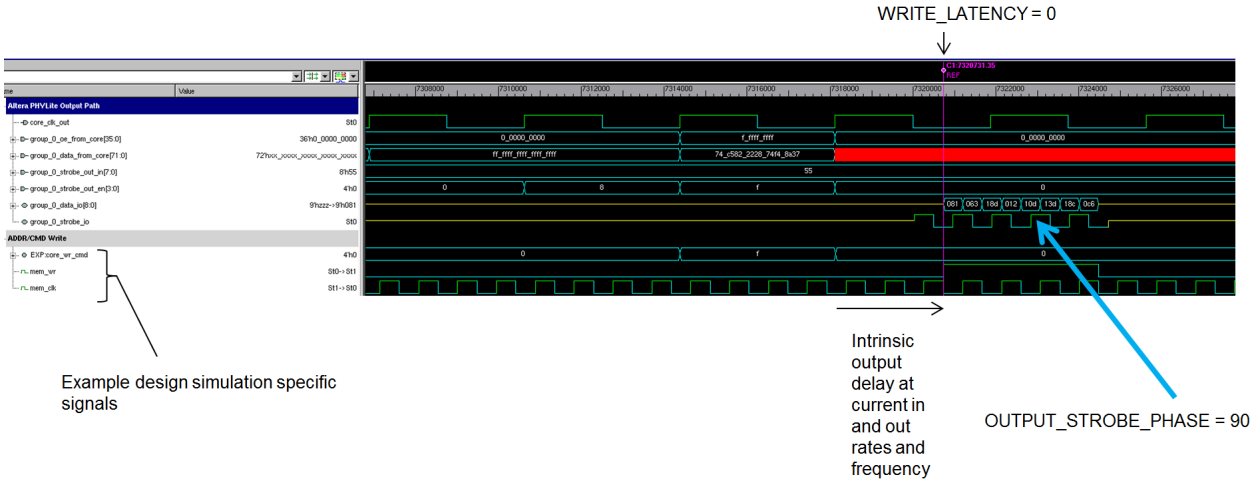
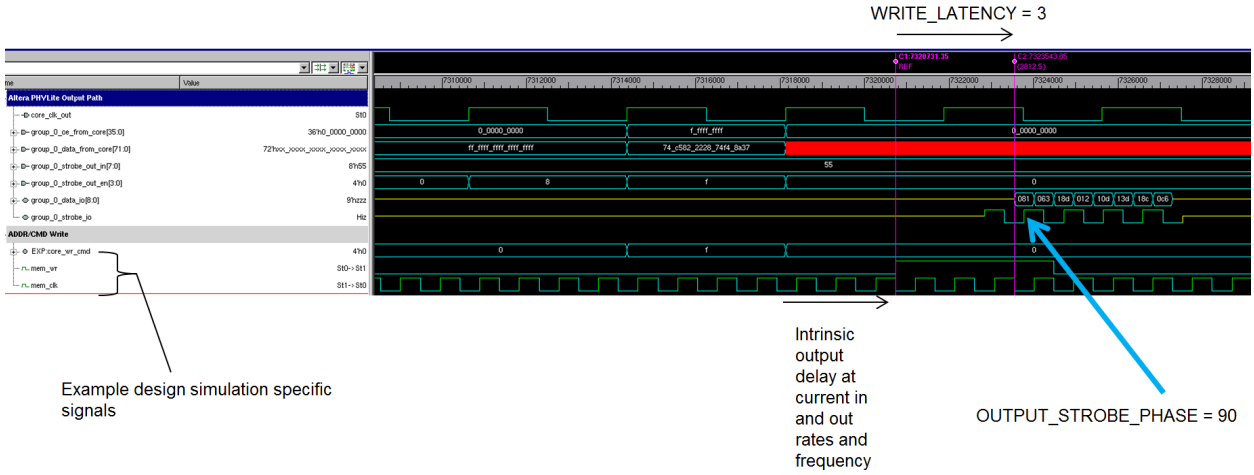

WRITE_LATENCY = 0
Example design simulation specific signals
Intrinsic output delay at current in and out rates and frequency
OUTPUT_STROBE_PHASE = 90

Figure 7: Output Path — Write Latency 3


WRITE_LATENCY = 3
Example design simulation specific signals
Intrinsic output delay at current in and out rates and frequency
OUTPUT_STROBE_PHASE = 90

Related Information

- **Output Path Signals** on page 38
  For more information about output path signals
- **Dynamic Reconfiguration** on page 20

Output Path Data Alignment

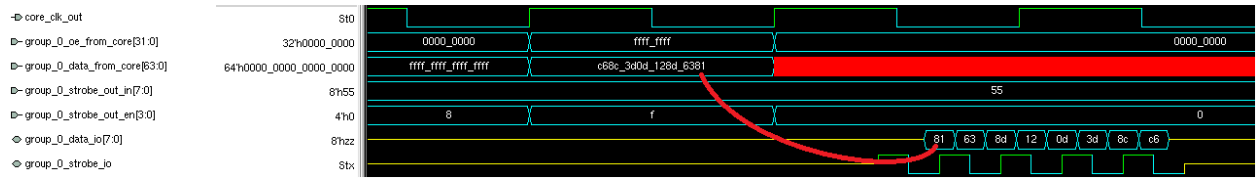
The `data_from_core` and `oe_from_core` signals are arranged in time slices, which are broken down into the individual pins in the group. The first time slice is on the LSBs of the buses, which matches the Altera PHY interface (AFI) bus ordering of the Arria 10 External Memory Interfaces IP core.

Example of time slices with individual pins correlation:

`{...,time2,time1,time0}`

Where `time0` = `{...,pin1,pin0}`

Figure 8: Example Output for Quarter Rate DDR



Related Information

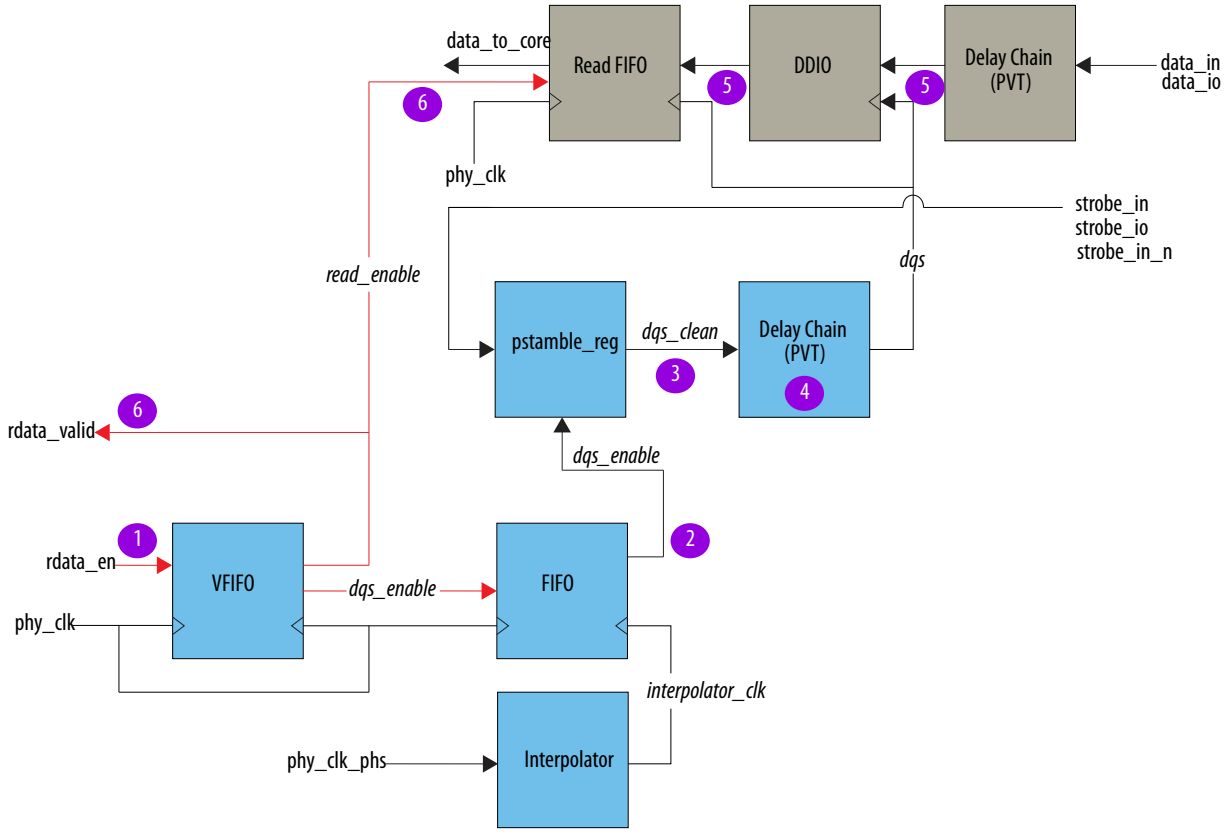
Dynamic Reconfiguration on page 20

External Memory Interface Handbook
For more information about the AFI 3.0 specification

Input Path

Figure 9: Input Path

This figure shows the input path of the IP core.



The input path of the IP core consists of a data path, a strobe path, and read enable path.

Table 4: Blocks in Data, Strobe, and Read Enable Paths

This table lists the information about these paths.

Path	Description
Data Path	Consists of a PVT compensated delay chain, a DDIO and a read FIFO. • PVT compensated delay chain—Allows per-bit deskew. You can only control the PVT compensated delay chain over Avalon-MM interface. For more information, refer to **Dynamic Reconfiguration**. • DDIO and read FIFO—Responsible for deserialization with a factor of up to 8 (in DDR quarter-rate). The transfer between the DDIO and the read FIFO is a zero-cycle transfer. The IP core supports SDR input by dropping every other bit of data going to the core.
Strobe Path	Consists of pstamble_reg (a gating component) and a PVT compensated delay chain. • pstamble_reg—This gating circuitry ensures that only clock edges associated with valid input data are used. • PVT compensated delay chain—Provides a phase offset between the strobe and the data (for example, center aligning edge-aligned inputs).
Read Enable Path	Consists of VFIFO, FIFO, and an interpolator. • VFIFO—takes the `rdata_en` signal from the core and delays it separately for two outputs, one for the read enable on the read FIFO, and one for the strobe enable. These delays are calculated at generation time based on the read latency that you provide. Individual control is not necessary, but if you are modifying these delays you can do so individually using dynamic reconfiguration. • FIFO and interpolator—used for the strobe enable delay, the FIFO and interpolator are identical to the FIFO and interpolator circuitry in the output path. The FIFO and interpolator are configured to match the output delay for a group with no additional output delay (Write latency = `0`). During dynamic reconfiguration, the FIFO and interpolator can be used for fine grained control of the strobe enable signal. Both of these delays are controlled by the **Read latency** parameter for the group.

Table 5: Read Operation Sequence

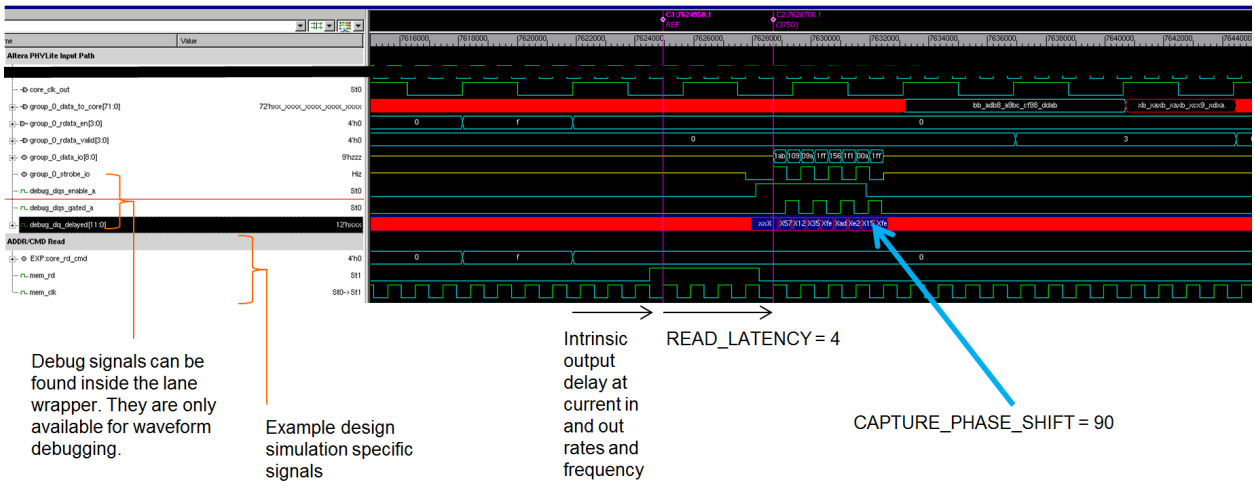
A read operation is performed as listed in this table.

Legend in Figure 9	Operation
1	The core asserts the `read_en` signal (and the external device is issued a read command)
2	The strobe enable is delayed through the two FIFOs by the programmed read latency (which should match the latency of the external device)
3	The strobe signal is ungated by the strobe enable signal as valid data enters the read path

Legend in Figure 9	Operation
4	The strobe is optionally delayed to create a phase offset between the strobe and the input data (for example, 90° phase shift for DDR center-alignment)
5	The data is clocked into the DDIO and read FIFO by the strobe
6	The VFIFO asserts the read enable on the read FIFO and the `rdata_valid` signal to the core simultaneously. This outputs the captured data and the associated valid signal to the core.

Figure 10: Input Path Waveform

This figure shows a waveform diagram of the input path.


Debug signals can be found inside the lane wrapper. They are only available for waveform debugging.
Example design simulation specific signals
Intrinsic output delay at current in and out rates and frequency
READ_LATENCY = 4
CAPTURE_PHASE_SHIFT = 90

Related Information

Input Path Signals on page 40
For more information about input path signals

Input Path Data Alignment

The bus ordering of `data_to_core`, `rdata_en`, and `rdata_valid` is identical to the ordering of the output path. That is, the LSBs of the bus hold the first time slice of data received.

The `rdata_valid` delay is always set by the IP core to match the `rdata_en` alignment. For example, quarter-rate delays are multiples of four external memory clock cycles (one quarter rate clock cycle).

Unaligned reads will result in unaligned `rdata_valid` and `data_to_core` with `data` and `valid` signals packed to the LSBs.

Figure 11: Example Input (Quarter Rate DDR) - Aligned

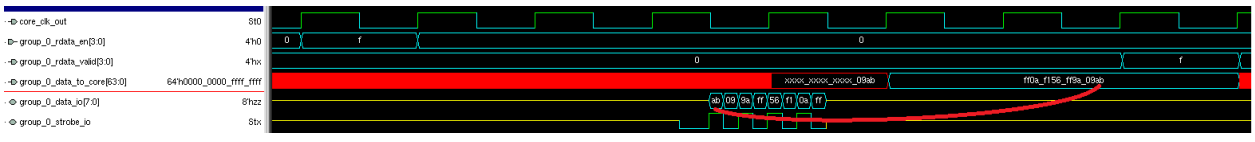
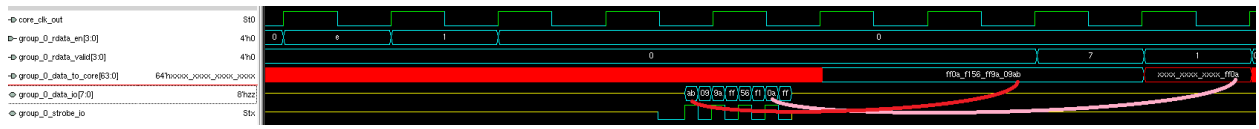


Figure 12: Example Input (Quarter Rate DDR) - Unaligned



I/O Standards

The Altera PHYLite for Parallel Interfaces IP core allows you to set I/O standards on the pins associated with the generated configuration. The I/O standard controls the available strobe configurations and OCT settings for all groups.

When you select an I/O standard in the I/O standard parameter, the reference clock assigns the I/O standard as a single-ended input. For a differential reference clock, override the single-ended Quartus II IP File (**.qip**) setting in the **.qsf**.

If you want to assign I/O standards manually at the system level (in the **.qsf**), then set the I/O standard to none, which will not output any I/O standard related **.qip** assignments from the IP generation.

Table 6: I/O Standards

I/O Standard	Valid Input Terminations (Ω) (1)	Valid Output Terminations (Ω)(1)	RZQ (Ω)	Differential/Complementary I/O Support
SSTL-12	60, 120	40, 60	240	Yes
SSTL-125	20, 30, 40, 60, 120	34, 40	240	Yes
SSTL-135	20, 30, 40, 60, 120	34, 40	240	Yes
SSTL-15	20, 30, 40, 60, 120	34, 40	240	Yes
SSTL-15 Class I	0, 50	0, 50	100	Yes
SSTL-15 Class II	0, 50	0, 25	100	Yes
SSTL-18 Class I	0, 50	0, 50	100	Yes
SSTL-18 Class II	0, 50	0, 25	100	Yes
1.2-V HSTL Class I	0, 50	0, 50	100	Yes
1.2-V HSTL Class II	0, 50	0, 25	100	Yes
1.5-V HSTL Class I	0, 50	0, 50	100	Yes
1.5-V HSTL Class II	0, 50	0, 25	100	Yes
1.8-V HSTL Class I	0, 50	0, 50	100	Yes
1.8-V HSTL Class II	0, 50	0, 25	100	Yes

(1) 0 is equivalent to none.

I/O Standard	Valid Input Terminations (Ω) [1]	Valid Output Terminations (Ω)[1]	RZQ (Ω)	Differential/Complementary I/O Support
1.2-V POD	34, 40, 48, 60, 80, 120, 240	34, 40, 48, 60	240	Yes
1.2-V	—	—	—	No
1.5-V	—	—	—	No
1.8-V	—	—	—	No

Input Buffer Reference Voltage (VREF)

The 1.2-V POD I/O standard allows a configurable VREF. By default, the externally provided VREF is used and using an internal VREF requires the following **.qsf** assignments:

```
set_instance_assignment -name VREF_MODE <mode> -to <pin_name>
```

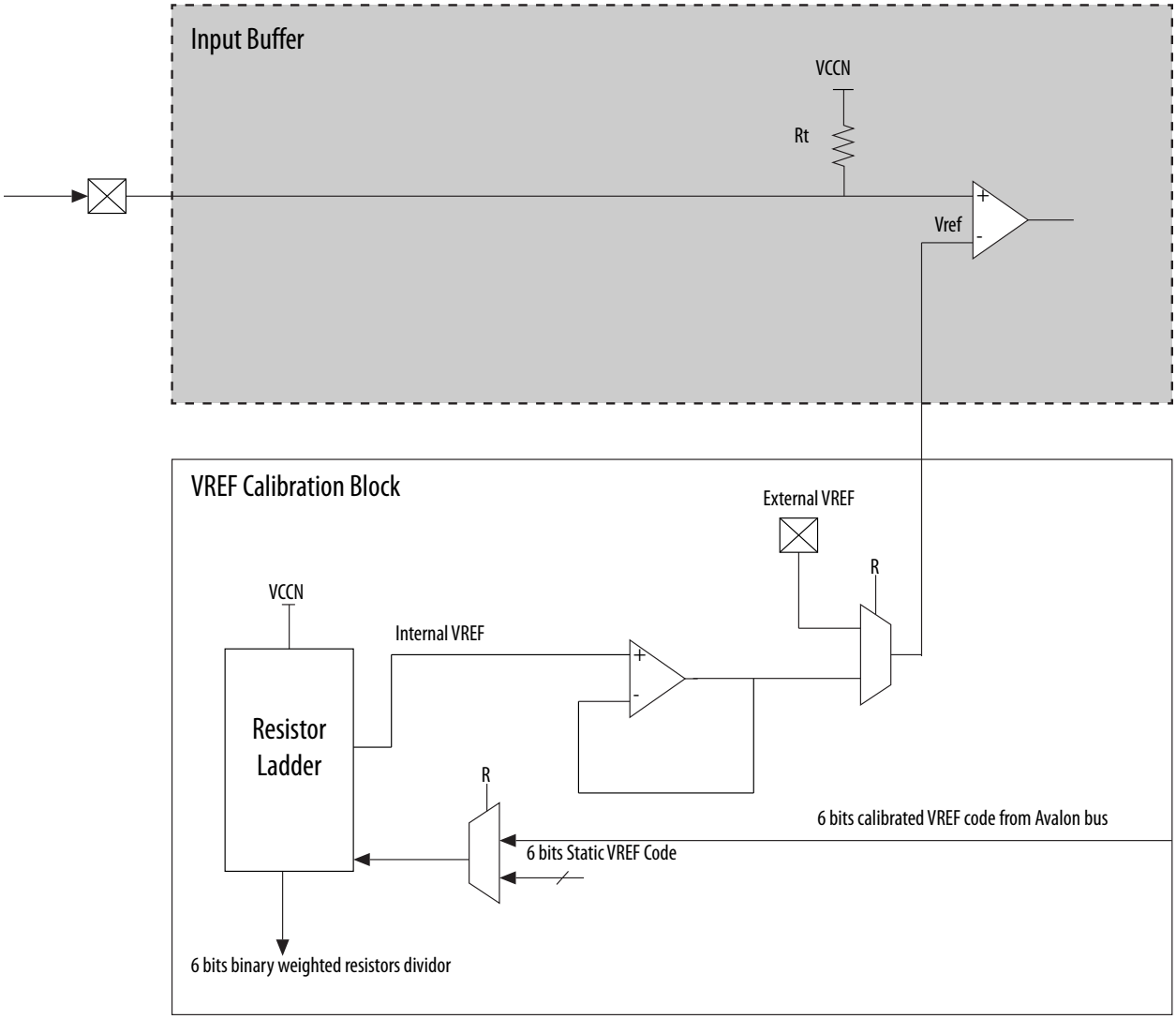
Note: The VREF settings are at the lane level, so all pins using a lane must have the same VREF settings (including GPIOs).

Table 7: VREF_MODE Description

VREF Mode	Description
EXTERNAL	Use the external VREF. This is the default.
CALIBRATED	Use internal VREF generated using VREF codes from the Avalon reconfiguration bus.
VCCIO_45	Use internal VREF generated using static VREF code. VREF is 45% of VCCN
VCCIO_50	Use internal VREF generated using static VREF code. VREF is 50% of VCCN
VCCIO_55	Use internal VREF generated using static VREF code. VREF is 55% of VCCN
VCCIO_65	Use internal VREF generated using static VREF code. VREF is 65% of VCCN
VCCIO_70	Use internal VREF generated using static VREF code. VREF is 70% of VCCN
VCCIO_75	Use internal VREF generated using static VREF code. VREF is 75% of VCCN

[1] 0 is equivalent to none.

Figure 13: VREF



Calibrated VREF Settings

Table 8: Calibrated VREF Settings

This table lists the calibrated VREF settings that you can set over the Avalon calibration bus.

avl_writedata[5:0]	% of VCCN
000000	60.00%
000001	60.64%
000010	61.28%
000011	61.92%
000100	62.56%

avl_writedata[5:0]	% of VCCN
000101	63.20%
000110	63.84%
000111	64.48%
001000	65.12%
001001	65.76%
001010	66.40%
001011	67.04%
001100	67.68%
001101	68.32%
001110	68.96%
001111	69.60%
010000	70.24%
010001	70.88%
010010	71.52%
010011	72.16%
010100	72.80%
010101	73.44%
010110	74.08%
010111	74.72%
011000	75.36%
011001	76.00%
011010	76.64%
011011	77.28%
011100	77.92%
011101	78.56%
011110	79.20%
011111	79.84%
100000	80.48%
100001	81.12%
100010	81.76%
100011	82.40%

avl_writedata[5:0]	% of VCCN
100100	83.04%
100101	83.68%
100110	84.32%
100111	84.96%
101000	85.60%
101001	86.24%
101010	86.88%
101011	87.52%
101100	88.16%
101101	88.80%
101110	89.44%
101111	90.08%
110000	90.72%
110001	91.36%
110010	92.00%
110011 -> 111111	Reserved

Related Information

Dynamic Reconfiguration on page 16
For more information on the Avalon bus usage

Placement Restrictions

Group Pin Placement

Place each group in the interface into a set of lanes in the same bank, the number of which depends on the number of pins used by the group. All groups in an interface must be placed across a contiguous set of banks.

Table 9: Group Pin Placement

Number of Pins in Group	Valid DQS Group in a Bank	Valid Indices in a Bank
1-12	DQS for X8/X9	{0-11}/{12-23}/{24-35}/{36-47}
13-24	DQS for X16/X18	{0-23}/{24-47}
24-48	DQS for X32/X36	{0-47}

Related Information

- **Device Pin-Out File**
  For specific DQS group numbers refer to the specific device Pin-Out file

Reference Clock

The reference clock must be placed on a clock input in one of the banks used by the interface. If the reference clock is used for multiple interfaces (consisting of a combination of EMIF and Altera PHYLite for Parallel Interfaces IPs), it can be placed in any bank used by any of the interfaces, but the banks for all interfaces must be contiguous.

Constraining Multiple Altera PHYLite for Parallel Interfaces to One I/O Bank

To constrain groups from separate Altera PHYLite for Parallel Interfaces IP core instances into the same I/O bank, the instances must share the same reference clock and reset sources, the same external memory frequencies and the same voltage settings.

Dynamic Reconfiguration

If you are using the dynamic reconfiguration feature, all interfaces of the Arria 10 External Memory Interfaces and Altera PHYLite for Parallel Interfaces IP cores in the same I/O column must share the `reset` signal. Multiple IP cores requiring Avalon core access require daisy chain connectivity.

Related Information

Daisy Chain on page 20
Describes the daisy chain connectivity

Timing

The Quartus II software version 14.1 generates the required timing constraints to analyze the timing of the Altera PHYLite for Parallel Interfaces IP core on the Arria 10 device.

Timing Components

Table 10: Timing Components

Circuit Category	Timing Paths	Source	Destination	Description
Source Synchronous and optionally calibrated (2)	Read Path	Memory Device	DQ Capture Register	Source synchronous timing paths—paths where clock and data signals are passed from the transmitting devices to the receiving devices. Optionally calibrated paths—paths with delay elements that are dynamically reconfigurable to achieve timing closure, especially at higher frequency, and to maximize the timing margins. You can calibrate these paths by implementing an algorithm and turning on the optional dynamic reconfiguration feature. An example of the calibrated path is the FPGA to memory device write path, in which you can dynamically reconfigure the delay elements to, for instance, compensate the skew due to process voltage temperature variation.
Source Synchronous and optionally calibrated (2)	Write Path	FPGA DQ/DQS	Memory Device	
Internal FPGA	Core to PHYLite Path	Core Registers	Write FIFO	The internal FPGA paths are paths in the FPGA fabric. The TimeQuest timing analyzer reports the corresponding timing margins.
Internal FPGA	PHYLite to Core	Read FIFO	Core Registers	

Timing Constraints and Files

To enable you to successfully timing constrain the Altera PHYLite for Parallel Interfaces IP core, the IP core generates a set of timing files. You can locate these timing files in the <*variation_name*> directory:

- <*variation_name*> **.sdc**
- <*variation_name*> **_ip_parameters.tcl**
- <*variation_name*> **_pin_map.tcl**

(2) Can be optionally calibrated by using dynamic reconfiguration.

<variation_name>.sdc

You can find the location of the <*variation_name*>**.sdc** file in the **.qip**, which is generated during the IP generation. The <*variation_name*>**.sdc** allows the Fitter to optimize timing margins with timing driven compilation and allows the TimeQuest timing analyzer to analyze the timing of your design.

The IP core uses <*variation_name*>**.sdc** for the following operations:

- Creating clocks on PLL inputs
- Creating generated clocks
- Calling `derive_clock_uncertainty`
- Creating `set_output_delay` and `set_input_delay` constraints to analyze the timing of the read and write paths

<variation_name>_ip_parameters.tcl

The <*variation_name*>**_ip_parameters.tcl** file lists the Altera PHYLite for Parallel Interfaces IP core parameters and is read by the <*variation_name*>**.sdc**.

<variation_name>_pin_map.tcl

The <*variation_name*>**_pin_map.tcl** is a TCL library of functions and procedures that <*variation_name*>**.sdc** uses.

Timing Analysis

Table 11: Timing Analysis

This table lists the timing analysis performed in the I/O and FPGA for the Altera PHYLite for Parallel Interfaces IP core.

Location	Description
I/O	The Altera PHYLite for Parallel Interfaces IP core generation creates the appropriate generated clock settings for the read strobe on the read path and the write strobe of the write path, according to their strobe type (singled-ended, complementary, or differential) and their interface type (SDR or DDR) in the following format: • Clock name for read strobe—`<pin_name>_IN`. • Clock name for the write path—`<pin_name>` for positive strobe. • Clock name for the write path—`<pin_name>_neg` for negative strobe. The `set_false_path`, `set_input_delay` and `set_output_delay` constraints are also generated to ensure proper timing analysis of the Altera PHYLite for Parallel Interfaces IP core.

Location	Description
FPGA	The Altera PHYLite for Parallel Interfaces IP core generation creates the clock settings for the user core clock and the periphery clock in the following formats: • user core clock—<*variation_name*>_usr_clk • periphery clock— <*variation_name*>_phy_clk* The user core clock is for user core logic and the periphery clock is the clock for the PHYLite periphery hardware. With these clock settings, the TimeQuest Timing Analyzer analyzes the timing of the Altera PHYLite for Parallel Interfaces IP core interface transfer and within core transfer correctly.

Timing Closure Guidelines

Timing Closure: Dynamic Reconfiguration

You can dynamically reconfigure the delay elements in the I/O to optimize process, voltage, temperature variations by implementing a calibration algorithm that modifies the input and output delays (refer to **Dynamic Reconfiguration** on page 20).

The SDC cuts the I/O transfer paths and you must verify the reconfiguration algorithm to ensure that your I/O transfers are working. The Quartus II software issues the following critical warning:

```
Dynamic Reconfiguration is ON but user has not set var(dynamic_reconfigura-
tion_algorithm_verified) to 1. Please set to 1 after calibration algorithm is
extensively verified.  I/O timing analysis may not represent the system.
```

After verifying the algorithm, you can disable the critical warning by editing the **.sdc** file and set the following variable to 1:

```
var(dynamic_reconfiguration_algorithm_verified)
```

Timing Closure: Non Edge-Aligned Input Data

If the input data is not edge-aligned, modify the timing settings of the group to match the system. Convert input strobe phase shift to nanosecond and subtract it from **Input Strobe Setup Delay Constrain** and **Input Strobe Hold Delay Constrain** parameters.

If the input data is center-aligned with the input strobe, subtract the 90° phase shift from the **Input Strobe Setup Delay Constrain** and **Input Strobe Hold Delay Constrain** parameters in the <*variation_name*>**.sdc**. For example, if the memory speed is 800 MHz and the value of the **Input Strobe Setup Delay Constrain** parameter is 0.1, change the value to 0.1-1.25*(90/360) = -0.2125.

Note: Ensure that you make the changes in the **Input Strobe Setup Delay Constrain** and **Input Strobe Hold Delay Constrain** parameters.

I/O Timing Violation

At high frequency configuration, it is difficult to achieve timing closure at I/O. Consider using the Arria 10 External Memory Interface IP core or the dynamic reconfiguration feature to calibrate the I/O path.

Related Information

Dynamic Reconfiguration on page 20
For more information about using the dynamic reconfiguration feature to calibrate the I/O path

Internal FPGA Path Timing Violation

If timing violations are reported at the internal FPGA paths (such as `<instance_name>_usr_clk` or `<instance_name>_phy_clk_*`), consider the following guidelines:

If setup time violation is reported, lower the clock rate of the user logic from full-rate to half-rate, or from half-rate to quarter-rate. This reduces the frequency requirement of the IP core-to-core data transfer.

If hold time violation is observed, overconstrain the hold uncertainty in the **.sdc** to force the Fitter to fix the hold time violation. Under normal circumstances, the Fitter should already attempt to avoid hold time violation. It is possible that the Fitter may think adding more delay to avoid hold time at the fast timing corner may cause setup time violation at the slow corner.

Dynamic Reconfiguration

Due to the asynchronous nature of the PHY, you must perform calibration to achieve timing closure at a high frequency. At a high level, calibration involves reconfiguring input and output delays in the PHY to align data and strobes. Enabling dynamic reconfiguration in the Altera PHYLite for Parallel Interfaces IP core provides allows you to modify these delays using an Avalon-MM interface.

RTL Connectivity

When generating the Altera PHYLite for Parallel Interfaces IP core with the dynamic reconfiguration feature enabled, the Altera PHYLite for Parallel Interfaces IP core exposes the Avalon-MM master and Avalon-MM slave interfaces. If the generated IP core is the only Altera PHYLite for Parallel Interfaces IP core (with dynamic reconfiguration) or Arria 10 External Memory Interfaces IP core in the I/O column, then only the slave interface needs to be used with a master in the core. Otherwise, both interfaces must be connected as described in the following section.

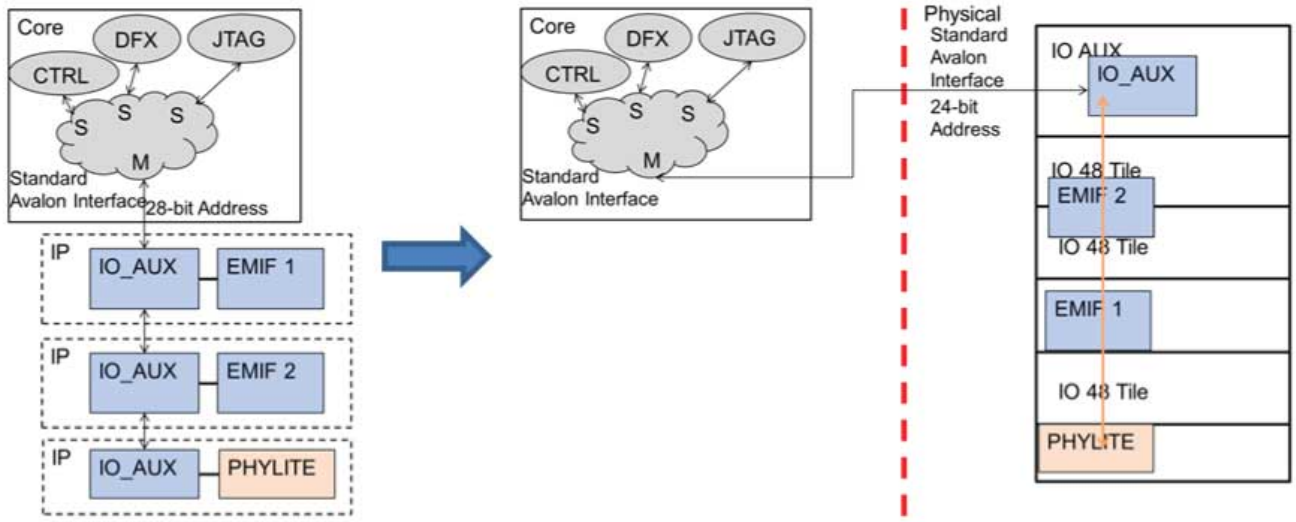
Daisy Chain

The I/O column provides a single physical Avalon-MM interface. All IP cores in the I/O column that require Avalon access from the core use the same physical Avalon-MM interface. The system level RTL for the column reflects this resource limitation by using a daisy chain to connect all dynamically reconfigurable IP cores in an I/O column.

The Altera PHYLite for Parallel Interfaces IP core exposes a 28-bit Avalon-MM address, where the top 4-bits are the ID of the interface to be addressed in the daisy chain. These bits are only required for the daisy chain arbitration in RTL simulation, so they are synthesized away during compilation. If only one interface is addressed from the core, it is sufficient to tie these bits off to the interface's ID.

Figure 14: Logical RTL View to Physical Column Placement

This figure shows an example of a daisy chain consisting of the Arria 10 External Memory Interfaces and Altera PHYLite for Parallel Interfaces IP cores before and after placement.



Notice that all core controllers must go through the arbitration logic that you created in the FPGA core logic to connect to an interface on the daisy chain. The end of the daisy chain should have its master output interface tied off.

Note: The prefit netlist of a design using the daisy chain will not simulate correctly due to the rearrangement of the Avalon address pins, which is done by the Fitter. The postfit netlist will properly simulate the merged I/O column.

Addressing

Each reconfigurable feature of the interface has an associated memory address. However, this address is placement dependent so addresses of the interface lanes, as well as the pins must be tracked in order to use the IP in a column that can be shared with other Altera PHYLite for Parallel Interfaces and the Arria 10 External Memory Interfaces IP cores, which also use the Avalon Bus.

Note: Addressing is done at the 32-bit word boundary; avl_address[1:0] = 00

Table 12: Address Map

Feature	Avalon Address R/W	Address CSR R	Control		Value
			Field	Range	
Pin Output Phase	`{id[3:0], 3'h4, lane_addr[7:0], pin{4:0], 8'D0}`	`{id[3:0], 3'h1, lane_addr[7:0], pin{4:0], 8'E8}`	Phase Value	12..0	Minimum Setting: Refer to **Table 13** Maximum Setting: Refer to **Table 13** Incremental Delay: 1/128th VCO clock period **Note:** The pin output phase switches from the CSR value to the Avalon value after the first Avalon write. It is only reset to the CSR value on a reset of the interface.
			Reserved	31..13	

Pin PVT Compensated Input Delay	`{id[3:0], 3'h4, lane_addr[7:0], 4'hC, lgc_sel[1:0], pin_off[2:0], 4'h0}`
	• lgc_sel[1:0] is:
	  • 2'b01 for pin <=5
	  • 2'b10 for pin > 5
	• pin_off[2:0] is:
	  • pin[2:0] for pin <= 5
	  • pin[2:0] - 3'h6 for pin > 5
		Not supported	Delay Value	8..0	Minimum Setting: 0
					Maximum Setting: 511 VCO clock periods
					Incremental Delay: 1/256th VCO clock period

			Reserved	11..9	
			Enable	12	
			Reserved	31..13	

Feature	Avalon Address R/W	Address CSR R	Control		Value
			Field	Range	
Strobe PVT compensated input delay [3]	{id[3:0], 3'h4, lane_addr[7:0], 4'hC, lgc_sel[1:0], 3'h6, 4'h0} • lgc_sel[1:0] = • 2'b01 for a • 2'b10 for b [4]	Not supported	Delay Value	9..0	Minimum Setting: 0 Maximum Setting: 1023 VCO clock periods Incremental Delay: 1/256th VCO clock period
			Reserved	11..10	
			Enable	12	
			Reserved	31..13	
Strobe enable phase [3]	{id[3:0], 3'h4, lane_addr[7:0], 4'hC, lgc_sel[1:0], 3'h7, 4'h0} • lgc_sel[1:0] is: • 2'b01 for a • 2'b10 for b [4]	• {id[3:0], 3'h1, lane_addr[7:0], 4'hC, 9'h194} • {id[3:0], 3'h1, lane_addr[7:0], 4'hC, 9'h198}	Phase Value	12..0	Minimum Setting: Refer to **Table 13** Maximum Setting: Refer to **Table 13** Incremental Delay: 1/128th VCO clock period
			Reserved	14..13	
			Enable	15	
			Reserved	31..16	
Strobe enable delay [3]	{id[3:0], 3'h4, lane_addr[7:0], 4'hC, 9'h008}	{id[3:0], 3'h1, lane_addr[7:0], 4'hC, 9'h1A8}	Delay Value	5..0	Minimum Setting: 0 external clock cycles Maximum Setting: 63 external memory clock cycles Incremental Delay: 1 external memory clock cycle
			Reserved	14..6	
			Enable	15	
			Reserved	31..16	

[3] Modifying these values must be done on all lanes in a group.
[4] Strobe logic b is only used by the negative pin of complementary strobes.

Feature	Avalon Address R/W	Address CSR R	Control		Value
			Field	Range	
Read valid delay [3]	`{id[3:0], 3'h4, lane_addr[7:0], 4'hC, 9'h00C}`	`{id[3:0], 3'h1, lane_addr[7:0], 4'hC, 9'h1A4}`	Delay Value	6..0	Minimum Setting: 0 external clock cycles Maximum Setting: 127 external memory clock cycles Incremental Delay: 1 external memory clock cycle
			Reserved	14..7	
			Enable	15	
			Reserved	31..16	
Internal VREF Code	`{id[3:0], 3'h4, lane_addr[7:0], 4'hC, 9'h014}`	Not supported	VREF Code	5..0	Refer to **Table 8**
			Reserved [5]	31..6[5]	

Where:

- `id[3:0]` refers to the **Interface ID** parameter
- `lane_addr[7:0]` refers to the address of a given lane in an interface. This is set by the Fitter and can be queried in the parameter table as described in the **Address Look-Up** on page 25.
- `pin[4:0]` refers to the physical location of the pin in a lane. A pin location is either determined by the Fitter or through a **.qsf** assignment and can be queried in the parameter table as described in the **Address Look-Up** on page 25.

Output and Strobe Enable Minimum and Maximum Phase Settings

When dynamically reconfiguring the interpolator phase settings, the values must be kept within the ranges below to ensure proper operation of the circuitry.

Table 13: Output and Strobe Enable Minimum and Maximum Phase Settings

VCO Multiplication Factor	Core Rate	Minimum Interpolator Phase	Maximum Interpolator Phase
1	Full	0x100	0xA80
	Half	0x280	0xBC0
	Quarter	0x180	0xA00
2	Full	0x180	0xFFF
	Half	0x100	0xFFF
	Quarter	0x380	0xFFF

[5] Reserved bit ranges must be zero

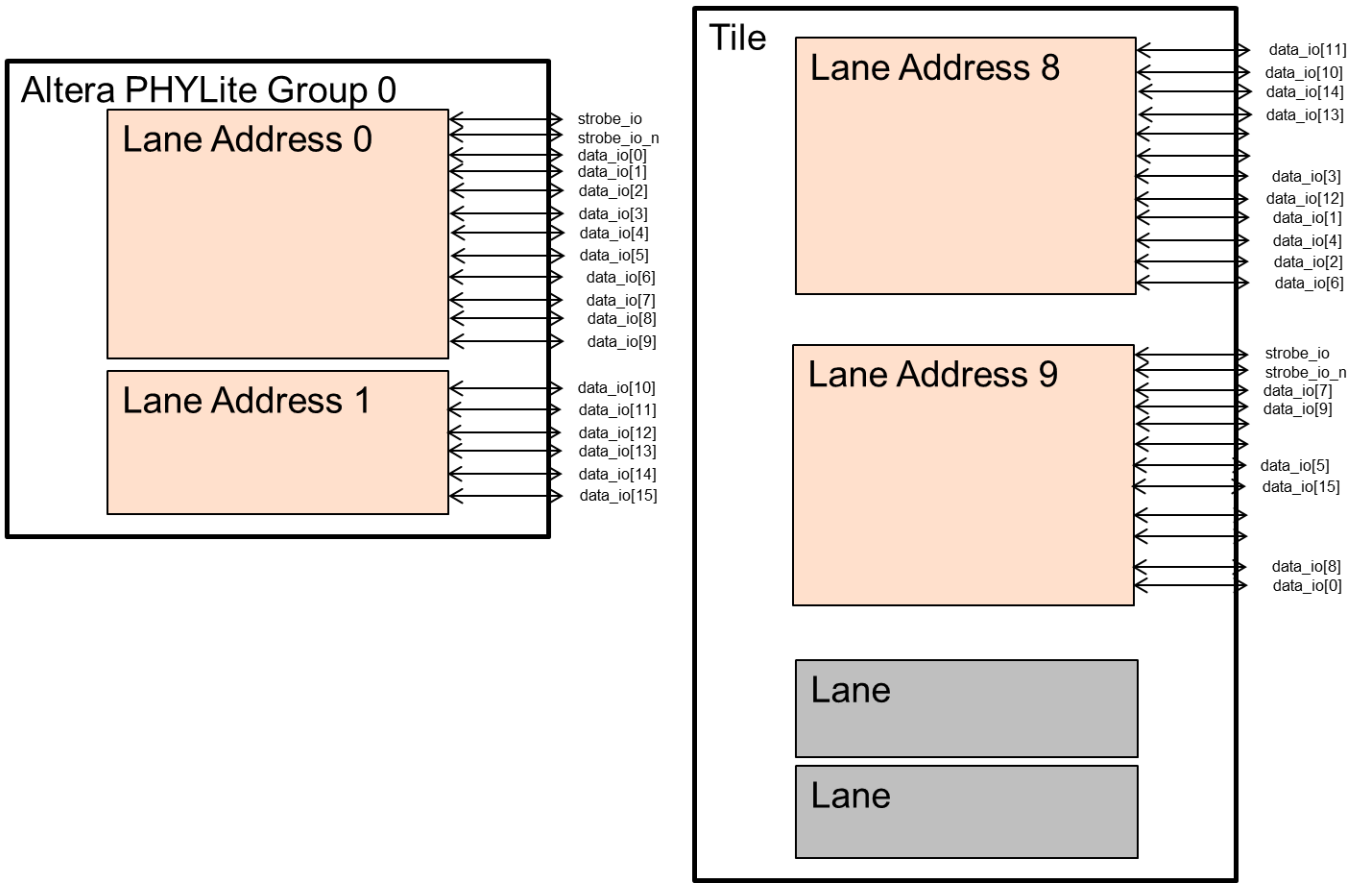
VCO Multiplication Factor	Core Rate	Minimum Interpolator Phase	Maximum Interpolator Phase
4	Full	0x200	0xFFF
	Half	0x100	0xFFF
	Quarter	0x280	0xFFF
8	Full	0x200	0xFFF
	Half	0x000	0xFFF
	Quarter	0x380	0xFFF

Address Look-Up

You must know the lane addresses and the pin placement to address an interface correctly. Because these values are placement dependent, these values will be different before and after placement. The Altera PHYLite for Parallel Interfaces IP core is generated as if the IP core is the only IP core in a column, with lane addresses starting from 0. If the IP core is placed in a column containing Arria 10 External Memory Interfaces or Altera PHYLite for Parallel Interfaces IP cores (with dynamic reconfiguration), then the addressing of the I/O lanes in the interfaces must be modified to avoid conflicts. A pin can also be moved into any lane within a group. In general, even if the Altera PHYLite for Parallel Interfaces IP core interface is the only IP in the column, the Fitter will still modify the addresses.

Figure 15: Lane and Pin Placement Dependent Addresses

This figure shows an example of a placed group with two lanes, 16 data pins and a differential strobe.



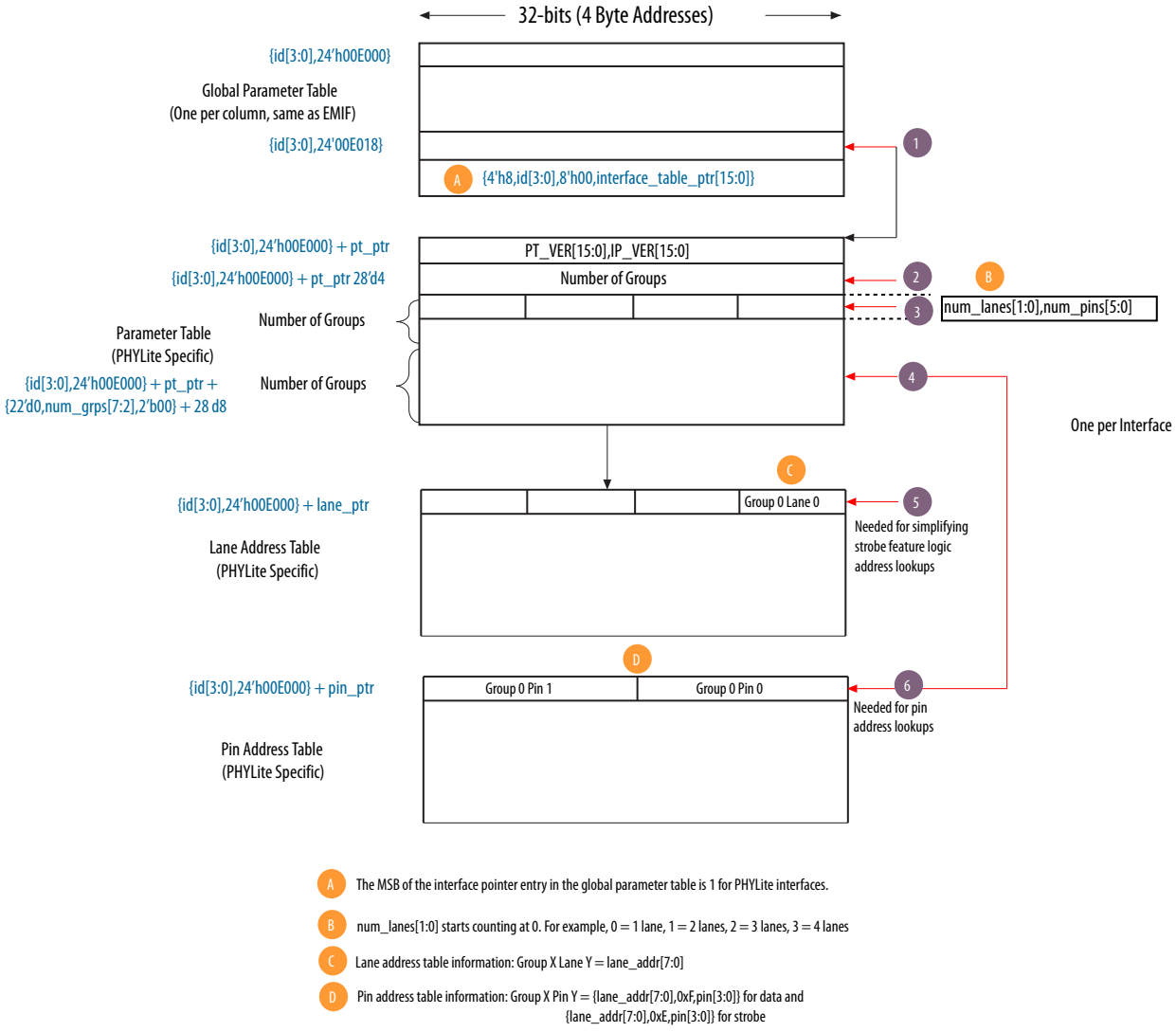
In order to provide a unified way to look up reconfigurable feature addresses for a specific interface both before and after placement, the address information is stored in memory in the I/O column. This memory is addressable over the same Avalon-MM bus as is used for feature reconfiguration.

Table 14: Memory Look Up Components

This table lists the two main components to the memory look-up.

Component	Description
Global parameter table	Stores pointers to the individual interface parameter tables. The global parameter table lists all interfaces in the column (both the Arria 10 External Memory Interfaces and Altera PHYLite for Parallel Interfaces IP cores).
Set of individual interface parameter tables	Contain interface specific information. This is where pin and lane level address look-ups are performed.

Figure 16: Memory Overview



The Parameter table look-ups are used as follows (the sequence corresponds to the sequence in **Figure 16**):

Table 15: Parameter Table Lookup Operation Sequence

Legend in Figure 16	Description
1	Search for Interface Parameter Table in Global Parameter Table (cache once per interface) • {id{3:0],24'h00E000} + 28'h18 to {id{3:0],24'h00E0000} + 28'h2C • 1 to 11 look-ups

Legend in Figure 16	Description
2	Retrieve number of groups in the interface (cache once per interface) • {id[3:0],24'h00E000} + {4'h0,pt_ptr[23:0]} + 4'h4 • You can skip this if it saved in the core during compilation (for example, hard coded in RTL logic)
3	Retrieve group information (cache once per group) • {id[3:0],24'h00E000} + {4'h0,pt_ptr[23:0]} + 24'h4 + grp_num • Not always necessary
4	Retrieve Lane/Pin Address Offsets for group (cache once per group) • {id[3:0],24'h00E000} + pt_ptr + {22'd0,num_grps[7:2],2'b00} + 28'd8
5	Perform lane/pin address translation (cache once per pin) • {id[3:0],24'h00E000} + {12'h000,lane_ptr[15:0]} + lane_num • {id[3:0],24'h00E000} + {12'h000,pin_ptr[15:0]} + {17'h0,pin_num[5:0], 1'b0}
6	Read/Write Avalon Calibration Bus • {id[3:0],24'h800000} + read_from_step_4 + intra_lane_addr

Caching look-ups 1-4 (8-bytes of information) allows for pin and lane translations in one look-up.

Strobes

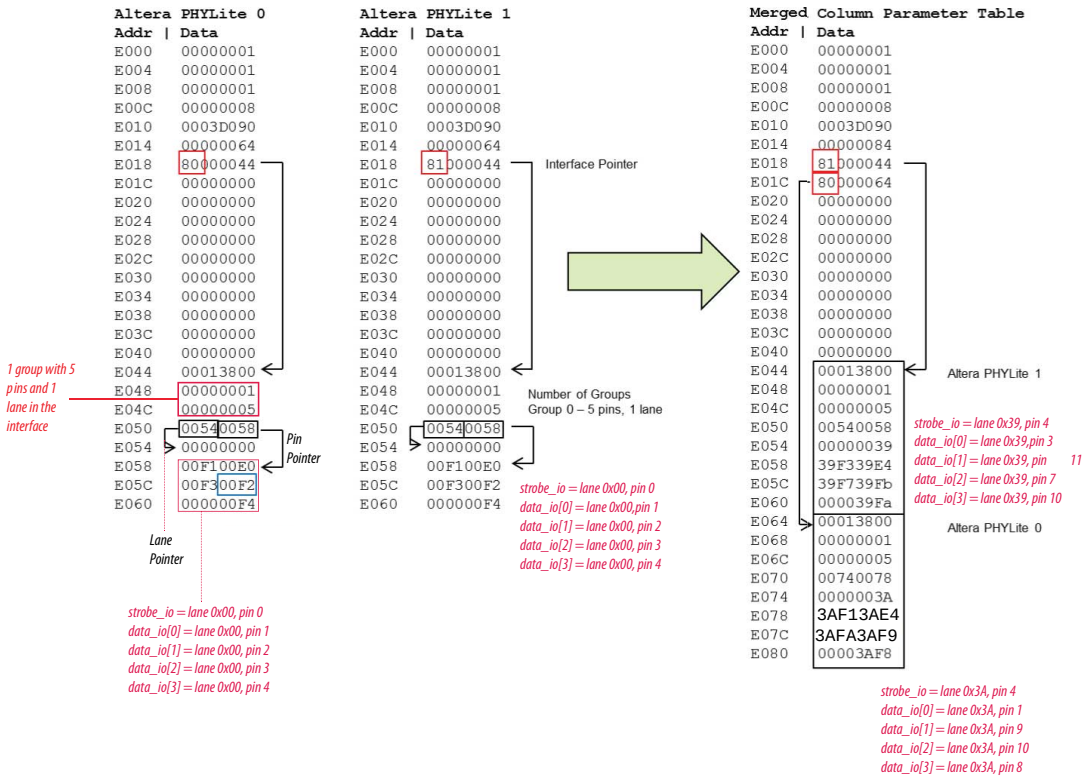
The first pins listed in the pin address look-up table are the strobes. They are also identified by `bits[7:4] = 0xE`. For separate strobes, the input strobe is always first. For differential and complementary strobes, the positive pin is the lower index.

Note: The output phase of differential strobes can be modified by writing to either the positive or negative pin. Only one write is necessary. This is also the case for output only complementary strobes.

Parameter Table Example

Figure 17: Parameter Table Example

This figure shows an example of a design containing two Altera PHYLite interfaces, each with one bidirectional group composed of 4 data bits and one strobe. Both interfaces are in the same I/O column and therefore their tables must be merged.



Note: Note there is no guarantee of the ordering of the interface parameter tables in the merged table, so a specific interface will have to be searched for.

For more information about the contents of the parameter table, refer to **Figure 16**.

Example Design Avalon Controller

An addressing operation can be complicated and error prone. Accidentally addressing the wrong interface of the IP cores may result in debugging runtime errors difficulties. Therefore, the example design provides an Avalon controller to simplify the dynamic control of an interface. Altera recommends you to simply integrate the provided example controller into a dynamic reconfiguration design.

Figure 18: Avalon Controller

The input interface is as follows:

```
avl_in_address[31:0] =
{8'h00,interface_id[3:0],grp[4:0],pin[5:0],csr[0],register[7:0]}
```

Note: There is no look-up stage here. All necessary data is automatically looked-up and cached by the Avalon controller.

Note: A single controller can support multiple interfaces in an I/O column.

Table 16: Avalon Controller Registers

This table lists the available registers in the Avalon controller. For more information, refer to **Table 12**.

Register[7:0]	Pin[5:0]	Csr[0]	Avalon Register Access Type	CSR Register Access Type	R/W Data on avl_readdata/avl_writedata
AVL_CTRL_REG_NUM_GROUPS	0	0: Access to Avalon register.	R	N/A	{24'h000000,num_grps[7:0]}
AVL_CTRL_REG_GROUP_INFO	0	0: Access to Avalon register.	R	N/A	{16'h0000,num_lanes[7:0],num_pins[7:0]}
AVL_CTRL_REG_IDELAY	0-47	0: Access to Avalon register.	R/W	N/A	{23'h000000,dq_delay[8:0]}
AVL_CTRL_REG_ODELAY	0-47	0: Access to Avalon register. 1: Access to CSR register. Only read operation is allowed.	R/W	R	{19'h00000,output_phase[12:0]}

Register[7:0]	Pin[5:0]	Csr[0]	Avalon Register Access Type	CSR Register Access Type	R/W Data on avl_readdata/avl_writedata
AVL_CTRL_REG_DQS_DELAY	0: DQSA 1: DQSB (6)	0: Access to Avalon register.	R/W	N/A	{22'h000000,dqs_delay[9:0]}
AVL_CTRL_REG_DQS_EN_DELAY	0	0: Access to Avalon register. 1: Access to CSR register. Only read operation is allowed.	R/W	R	{26'h0000000,dqs_en_delay[5:0]}
AVL_CTRL_REG_DQS_EN_PHASE_SHIFT	0: DQSA 1: DQSB (6)	0: Access to Avalon register. 1: Access to CSR register. Only read operation is allowed.	R/W	R	{19'h00000,phase[12:0]}
AVL_CTRL_REG_RD_VALID_DELAY	0	0: Access to Avalon register. 1: Access to CSR register. Only read operation is allowed.	R/W	R	{25'h0000000,rd_vld_delay[6:0]}

The `interface_id[3:0]` and `grp[4:0]` components of the input address are always used.

Note: VREF reconfiguration is not currently supported by the example design Avalon controller.

(6) Strobe logic B is only used by the negative pin of complementary strobes

Calibration Guidelines

The Altera PHYLite for Parallel Interfaces IP core allows you to dynamically reconfigure the features of the interface. However, performing calibration is an application specific process. This section provides some general guidelines for calibrating the Arria 10 I/O architecture.

Strobe Enable Windowing

The read FIFO has the read pointer reset when reads are far apart (80 core clock cycles). However, the data inside the FIFO is not cleared. Therefore, an alternating pattern should be used to find the end to the strobe enable window to avoid erroneous correct reads due to stale data in the FIFO.

The strobe enable turns itself off on the last negative edge of the strobe. Therefore, while finding the enable window, use extra dummy pulses (either extended strobe or reads from memory without asserting the `rdata_en` signal) to clear the strobe enable.

Altera PHYLite for Parallel Interfaces IP Core Reference

Parameter Settings

Table 17: Altera PHYLite for Parallel Interfaces IP Core Parameter Settings

This table lists the parameter settings for the Altera PHYLite for Parallel Interfaces IP core.

GUI Name	Values	Description
Parameter		
Number of groups	1 to 18	Number of data/strobe groups in the interface. The value is set to 1 by default.
General Tab- these parameters are set on a per interface basis		
Clocks		
Memory clock frequency	100 MHz - 1333.333 MHz	External memory clock frequency. The value is set to 533 MHz by default. **Note:** To achieve timing closure at 800 MHz and above, use dynamic reconfiguration to calibrate the interface.
Use core PLL reference clock connection	—	Enables reference clock source supplied by the core. By default, the PLL reference clock source is a dedicated clock pin.
Use recommended PLL reference clock frequency	—	If you want to calculate the PLL reference clock frequency automatically for best performance, then turn on this option. If you want to specify your own PLL reference clock frequency, then turn off this option. This option is enabled by default.

GUI Name	Values	Description
PLL reference clock frequency	Dependent on desired memory clock frequency	PLL reference clock frequency. You must feed a clock of this frequency to the PLL reference clock input of the memory interface. The default value is dependent on recommended clock frequency and user clock rate. **Note:** There is no minimum range, but the maximum output frequency is 1600 MHz limited by the clock network. The minimum range for the `ref_clk` signal is 10 MHz but the maximum is dependent on the speed grade.
Clock rate of user logic	Full, Half, Quarter	Determines the clock frequency of user logic in relation to the memory clock frequency. For example, if the memory clock sent from the FPGA to the memory device is toggling at 800 MHz, a "Quarter rate" interface means that the user logic in the FPGA runs at 200 MHz. The value is set to **Quarter** by default.
Specify additional output clocks based on existing PLL	—	Exposes additional output clocks from the existing PLL.
Number of additional clocks	0 to 4	Specifies the number of additional clocks to be exposed.
Desired Frequency	—	Specifies the output clock frequency of the corresponding output clock port, `outclk[]`, in MHz. The default value is 100.0 MHz. The minimum and maximum values depend on the device used. The PLL only reads the numerals in the first six decimal places.
Actual Frequency	—	Allows you to select the actual output clock frequency from a list of achievable frequencies. The default value is the closest achievable frequency to the desired frequency.
Phase Shift units	ps or degrees	Specifies the phase shift unit for the corresponding output clock port, `outclk[]` , in picoseconds (ps) or degrees.
Phase Shift	—	Specifies the requested value for the phase shift. The default value is 0 ps.
Actual Phase Shift	—	Allows you to select the actual phase shift from a list of achievable phase shift values. The default value is the closest achievable phase shift to the desired phase shift.

GUI Name	Values	Description
Desired Duty Cycle	0.0–100.0	Specifies the requested value for the duty cycle. The default value is 50.0%
Actual Duty Cycle	—	Allows you to select the actual duty cycle from a list of achievable duty cycle values. The default value is the closest achievable duty cycle to the desired duty cycle.
Dynamic Reconfiguration		
Use dynamic reconfiguration	—	Exposes an Avalon-MM interface, allowing you to control the configuration of the Altera PHYLite for Parallel Interfaces IP core settings.
Interface ID	0	The ID used to identify this interface in the column over the Avalon-MM bus.
I/O Settings		
I/O standard	SSTL-12 SSTL-125 SSTL-135 SSTL-15 SSTL-15 Class I SSTL-15 Class II SSTL-18 Class I SSTL-18 Class II 1.2-V-HSTL Class I 1.2-V-HSTL Class II 1.5-V-HSTL Class I 1.5-V-HSTL Class II 1.8-V-HSTL Class I 1.8-V-HSTL Class II 1.2-V POD 1.2-V 1.5-V 1.8-V None	Specifies the I/O standard of the interface's strobe and data pins written to the **.qip** of the IP instance. When you choose **None**, the I/O standard is unspecified in the generated IP.
Group <x> - these parameters are set on a per group basis		

GUI Name	Values	Description
Group <x> Pin Settings		
Pin type	Input, Output, Bidirectional	Direction of data pins. This value is set to Bidirectional by default.
Pin width	1 to 48	Number of pins in this data/strobe group. The value is set to 9 by default. A data width of 48 is only achievable if no strobe is used in the group. The number of strobes is controlled by the **Use output strobe**, **Strobe configuration** and **Use separate capture strobe** parameters.
DDR/SDR	DDR, SDR	Double/single data rate. The value is set to DDR by default.
Group <x> Input Path Settings		
Read latency	1 to 63 external interface clock cycles	Expected read latency of the external device in memory clock cycles. The value is set to 4 by default. For example, a design with an external clock frequency of 533 MHz in half-rate has a valid read latency range of 5 to 63 external interface clock cycles. Refer to **Table 18**.
Swap capture strobe polarity	—	Internally swap the negative and positive capture strobe input pins. This feature is only available for complementary strobe configurations.
Capture strobe phase shift	0,45,90,135, 180	Phase shift of input strobe relative to input data. The value is set to 90° by default.
Group <x> Output Path Settings		
Write latency	0 to 3 (maximum value is dependent on the rate)	Additional delay added to the output data in memory clock cycles.
Use output strobe	—	Use an output strobe. This option is enabled by default.
Output strobe phase	0,45,90,135,180	Phase shift of the output strobe relative to the output data. The value is set to 90° by default.
Group <x> General Data Settings		
Data configuration	Single ended, Differential	Selects the type of data. Single ended data type uses one pin. Differential data type uses 2 pins. By default, the value is set to single ended.

GUI Name	Values	Description
Group <x> General Strobe Settings		
Strobe configuration	Single ended, Differential, Complementary	Select the type of strobe. A single ended strobe uses one pin, which will reduce the maximum possible number of data pins in the group to 47. Differential/complementary strobe types use 2 pins, which will reduce the maximum possible number of data pins in the group to 46. The value is set to **Single ended** by default. **Note:** The differential strobe configuration uses a differential input buffer, which produces a single clock for the capture DDIO and read FIFO. The complementary strobe configuration uses two single-ended input buffers and clocks the data into the capture DDIO and read FIFO using both clocks (as required by protocols such as QDRII). The output path functionality is the same.
Use separate strobes	—	Separate the bidirectional strobe into input and output strobe pins. Using separate strobes is only available for a bidirectional data group with the output strobe enabled.
Group <x> OCT Settings		
Use Default OCT Values	—	Use default OCT values based on the **I/O standard** parameter setting.
Input OCT Value	No termination, 50 ohm with calibration	Specifies the group's data and strobe input termination values to be written to the **.qip** of the IP instance. The list of legal values is dependent on the **I/O standard** parameter setting. Refer to Table 6. This option is available when the **Use Default OCT Values** option is disabled.

GUI Name	Values	Description
Output OCT Value	No termination, 50 ohm with calibration	Specifies the group's data and strobe input termination values to be written to the **.qip** of the IP instance. The list of legal values is dependent on the **I/O standard** parameter setting. Refer to **Table 6**. This option is available when the **Use Default OCT Values** option is disabled.
Group <x> Timing Settings		
Generate Input Delay Constraints for this group	—	Instructs SDC to generate set_input_delay constraints for this group.
Input Strobe Setup Delay Constrain	Constraint in ns	Specifies the group's input setup delay constraint against the input strobe.
Input Strobe Hold Delay Constrain	Constraint in ns	Specifies the group's input hold delay constraint against the input strobe.
Generate Output Delay Constraints for this group	—	Instructs SDC to generate set_output_delay constraints for this group.
Output Strobe Setup Delay Constrain	Constraint in ns	Specifies the group's output setup delay constraint against the input strobe.
Output Strobe Hold Delay Constrain	Constraint in ns	Specifies the group's output hold delay constraint against the input strobe.

Read Latencies

Table 18: Read Latencies

This table list the read latencies.

VCO Frequency Multiplication Factor	Core Clock Rate Setting		
	Full-Rate	Half-Rate	Quarter-Rate
1	4	5	7
2	4	5	7
4	3	4	7
8	3	4	7

Related Information

VCO Frequency Multiplication Factor on page 4
Provides information for going from speed grade and external memory frequency to VCO multiplication factor

Signals

Clock and Reset Interface Signals

Table 19: Clock and Reset Interface Signals

Signal Name	Direction	Width	Description
ref_clk	Input	1	Reference clock for the PLL. The reference clock must be the same frequency as specified in the parameter.
reset_n	Input	1	Resets the interface. This signal is asynchronous.
interface_locked	Output	1	Interface locked signal from core. This signal indicates that the PLL and PHY circuitry are locked.
core_clk_out	Output	1	Use this core clock in the core-to-periphery transfer of soft logic data and control signals. The core_clk_out frequency depends on the interface frequency and clock rate of user logic parameter.

Output Path Signals

Table 20: Output Path Signals

Output path signals are signals that are available when you set the **Pin Type** parameter to either **Output** or **Bidirectional**.

Signal Name	Direction	Width	Description
oe_from_core	Input	Quarter-rate: 4 x PIN_WIDTH Half-rate: 2 x PIN_WIDTH Full-rate: 1 x PIN_WIDTH	Core rate data enable to be output. Synchronous to the core_clk output from the IP core.
data_from_core	Input	Quarter rate-DDR: 8 x PIN_WIDTH Half-rate DDR: 4 x PIN_WIDTH Full-rate DDR: 2 x PIN_WIDTH Quarter-rate SDR: 4 x PIN_WIDTH Half-rate SDR: 2 x PIN_WIDTH Full-rate SDR: 1 x PIN_WIDTH	Core rate data to be output. Synchronous to the core_clk output from the IP core.
strobe_out_in	Input	Quarter-rate: 8 Half-rate: 4 Full-rate: 2	Strobe pattern to be output. Synchronous to the core_clk output from the IP core. **Note:** This path is always DDR.

Signal Name	Direction	Width	Description
strobe_out_en	Input	Quarter-rate: 4 Half-rate: 2 Full-rate: 1	Enable output strobe. Synchronous to the core_clk output from the IP core.
data_out/data_io	Output/ Bidirectional	• 1 to 48 if data configuration is Single Ended • 1 to 24 if data configuration is Differential	Data output to pin. Synchronous to the strobe_out or strobe_io output from the IP core. If the **Pin Type** parameter is set to **Output**, the data_out signals are used. If the **Pin Type** parameter is set to **Bidirectional**, the data_io signals are used.
data_out_n/data_io_n	Output/ Bidirectional	1 to 24	Negative data output from pin enabled when data configuration is set to **Differential**. Data is synchronous to the strobe_out or strobe_io output from the IP core. If the pin type is set to **Output**, the data_out_n ports are used. If the pin type is set to bidirectional, the data_io_n ports are used.
strobe_out/ strobe_io	Output/ Bidirectional	1	Positive output strobe to pin. If the Pin Type is set to **Output**, the strobe_out signal is used. If the Pin Type is set to **Bidirectional** the strobe_io signal is used. The **Use Separate Strobes** parameter forces the use of the strobe_out signal with a Bidirectional Pin Type.

Signal Name	Direction	Width	Description
strobe_out_n/ strobe_io_n	Output/ Bidirectional	1	Negative output strobe to pin. This is used if the **Strobe Configuration** is set to **Differential** or **Complementary**. If the **Pin Type** is set to **Output**, the strobe_out_n signal is used. If the **Pin Type** is set to **Bidirectional**, the strobe_io_n signal is used. The **Use Separate Strobes** parameter forces the use of the strobe_out_n signal with a Bidirectional Pin Type.

Input Path Signals

Table 21: Input Path Signals

Input path signals are signals that are available when you set the **Pin Type** parameter to **Input** or **Bidirectional**.

Signal Name	Direction	Width	Description
data_to_core	Output	Quarter-rate DDR: 8 x PIN_WIDTH Half-rate DDR: 4 x PIN_WIDTH Full-rate DDR: 2 x PIN_WIDTH Quarter-rate SDR: 4 x PIN_WIDTH Half-rate SDR: 2 x PIN_WIDTH Full-rate SDR: 1 x PIN_WIDTH	Valid on rdata_valid. Synchronous to the core_clk output from the IP core.
rdata_en	Input	Quarter-rate: 4 Half-rate: 2 Full-rate: 1	Held high for the number of expected read words after a read command. Synchronous to the core_clk output from the IP core.
rdata_valid	Output	Quarter-rate: 4 Half-rate: 2 Full-rate: 1	Delayed by READ_LATENCY with margin and aligned to the core clock rate. For example, in quarter-rate, the delay will be a multiple of 4 external clock cycles. Synchronous to the core_clk output from the IP core.

Signal Name	Direction	Width	Description
data_in/ data_io	Input/ Bidirectional	• 1 to 48 if data configuration is Single Ended • 1 to 24 if data configuration is Differential	Data input from pin. Synchronous to the strobe_in or strobe_io input. If the pin type is set to **Input**, the data_in ports are used. If the pin type is set to **bidirectional**, the data_io ports are used.
data_in_n/ data_io_n	Input/ Bidirectional	1 to 24	Negative data input from pin enabled when data configuration is set to **Differential**. Data is synchronous to the strobe_in or strobe_io input. If the pin type is set to **Input**, the data_in_n ports are used. If the pin type is set to **bidirectional**, the data_io_n ports are used.
strobe_in/ strobe_io	Input/ Bidirectional	1	Positive strobe from pin. If the pin type is set to **Input**, the strobe_in signal is used. If the pin type is set to **Bidirectional**, the strobe_io signal is used.
strobe_in_n/ strobe_io_n	Input/ Bidirectional	1	Negative strobe from pin. This is used if the **Strobe Configuration** parameter is set to **Differential** or **Complementary**. If the pin type is set to **Input**, the strobe_in_n signal is used. If the pin type is set to **Bidirectional**, the strobe_io_n signal is used.

Avalon Configuration Bus Interface Signals

The Altera PHYLite for Parallel Interfaces IP core exposes the Avalon-MM slave and Avalon-MM master interfaces when you perform dynamic reconfiguration. Connect the Avalon-MM slave to either a master in the core or the master interface of either an Altera PHYLite for Parallel Interfaces IP core or the Arria 10 External Memory Interfaces IP core to be placed in the same column. You can only connect the master interface to the slave interface of an Altera PHYLite for Parallel Interfaces IP core or an Arria 10 External Memory Interfaces IP core to be placed in the same column.

Table 22: Avalon-MM Master Interface Signals

Signal Name	Direction	Width	Description
avl_clk	Input	1	Avalon interface clock.
avl_reset_n	Input	1	Reset input synchronous to avl_clk.

Signal Name	Direction	Width	Description
avl_read	Input	1	Read request from io_aux. This signal is synchronous to the avl_clk input.
avl_write	Input	1	Write request from io_aux. This signal is synchronous to the avl_clk input.
avl_byteenable	Input	4	Controls which bytes should be written on avl_writedata.
avl_writedata	Input	32	Write data from io_aux. This signal is synchronous to the avl_clk input.
avl_address	Input	28	Address from io_aux. This signal is synchronous to the avl_clk input.
avl_readdata	Output	32	Read data to io_aux. This signal is synchronous to the avl_clk input.
avl_writedata	Input	32	Write data from io_aux. This signal is synchronous to the avl_clk input.
avl_readdata_valid	Output	1	Indicates that read data has returned.
avl_waitrequest	Output	1	Stalls upstream logic when it is asserted.

Table 23: Avalon-MM Slave Interface Signals

Signal Name	Direction	Width	Description
avl_out_clk	Output	1	Connect this signal to the input Avalon interface of another Altera PHYLite for Parallel Interfaces IP core or the Arria 10 External Memory Interfaces IP.
avl_out_reset_n	Output	1	Connect this signal to the input Avalon interface of another Altera PHYLite for Parallel Interfaces IP core or the Arria 10 External Memory Interfaces IP.
avl_out_read	Output	1	Indicates read transaction.
avl_out_write	Output	1	Indicates write transaction.
avl_out_byteenable	Output	4	Controls which bytes should be written on avl_out_writedata.
avl_out_writedata	Output	32	The data packet associated with the write transaction.
avl_out_address	Output	28	Avalon address (in byte granularity). Value is identical to avl_address but with zeroes padded on the LSBs.
avl_out_readdata	Input	32	The data packet associated with avl_out_readdata_valid.
avl_out_readdata_valid	Input	1	Indicates that read data has returned.
avl_out_waitrequest	Input	1	Stalls upstream logic when it is asserted.

Related Information

Dynamic Reconfiguration on page 20
For more information about connecting these signals

Example Design

The Altera PHYLite for Parallel Interfaces IP core is able to generate an example design that matches the same configuration chosen for the IP. The example design is a simple design that does not target any specific application; however you can use the example design as a reference on how to instantiate the IP core and what behavior to expect in a simulation.

Note: The **.qsys** files are for internal use during example design generation only. You should not edit the files.

Generating Example Design

You can generate a example design by clicking **Example Design** in the IP Parameter Editor.

The software generates a user defined directory in which the example design files reside.

The **<instance>_example_design** directory contains two TCL scripts:

- `- make_qii_design.tcl`
- `- make_sim_design.tcl`

Generating Quartus Example Design

The `make_qii_design.tcl` generates a synthesizable example design along with a Quartus project, ready for compilation.

To generate synthesizable example design, run the following script at the end of IP generation:

```
quartus_sh -t make_qii_design.tcl
```

To specify an exact device to use, run the following script:

```
quartus_sh -t make_qii_design.tcl [device_name]
```

This script generates a **qii** directory containing a project called **ed_synth.qpf**. You can open and compile this project with the Quartus II software.

Generating Simulation Example Design

The `make_sim_design.tcl` generates a simulation example design along with tool-specific scripts to compile and elaborate the necessary files.

To generate a simulation example design for a Verilog or a mixed-language simulator, run the following script at the end of IP generation:

```
quartus_sh -t make_sim_design.tcl VERILOG
```

To generate simulation example design for a VHDL-only simulator, run the following script:

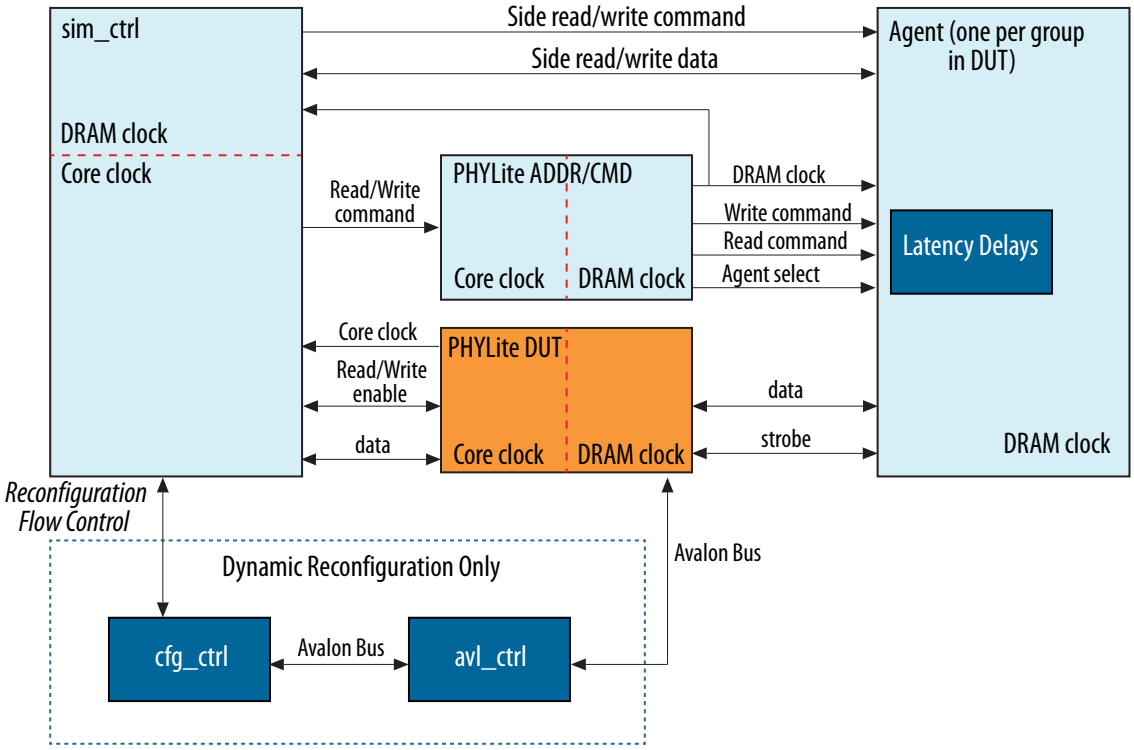
```
quartus_sh -t make_sim_design.tcl VHDL
```

This script generates a **sim** directory containing one subdirectory for each supported simulation tools. Each subdirectory contains the specific scripts to run simulation with the corresponding tool.

The simulation example design provides a generic example of the core and I/O connectivity for your IP configuration. Functionally, the simulation will iterate over each group in your configured IP and performs basic reads/writes to an associated agent (one per group) in the testbench. A simple one group Altera PHYLite instantiation in the testbench is used for basic address and command outputs to the agent. A side bus between the `sim_ctrl` and the agents is used to check that the reads and writes are valid.

Figure 19: High-Level View of the Simulation Example Design with One Group

This figure shows a high-level view of the simulation example design with one group.



Dynamic Reconfiguration Example Design

When you select the dynamic reconfiguration option, the example design introduces the `cfg_ctrl` and `avl_ctrl` blocks, which work with the `sim_ctrl` module to demonstrate the basic functionality of the Altera PHYLite IPs Avalon-MM based reconfiguration. The agent is also modified to insert delays on the data and clocks, which the new modules will compensate for.

Before sending test data, the `sim_ctrl` module first asks the `cfg_ctrl` to sweep for working delay values. While sweeping over the values, the `cfg_ctrl` module requests the `sim_ctrl` to perform writes and reads and return the results. The setting of the delays is simplified by the `avl_ctrl` module, which is described in detail in **Example Design Avalon Controller** on page 29.

NOTE: The `cfg_ctrl` module performs a simplistic reconfiguration of the interface that stops at the first working delay values. This works in simulation but will likely fail in a hardware scenario, as the initial working delay will be marginal. A robust calibration algorithm should sweep over the entire valid range of delays to choose the correct value for the application.

IP Migration for Arria V, Cyclone V, and Stratix V

In Arria 10 devices, you can instantiate the Altera PHYLite for Parallel Interfaces IP core on its own because the IP core contains the OCT and PLL. However, in Arria V, Cyclone V, and Stratix V devices, you must instantiate the ALTDQ_DQS2 IP core with the ALTERA_PLL, ALTDLL and ALTOCT IP cores.

Therefore, when migrating from the ALTDQ_DQS2 IP core to the Altera PHYLite for Parallel Interfaces IP core, you must:

- Configure the Altera PHYLite for Parallel Interfaces IP core settings.
- Manually remove the ALTERA_PLL, ALTOCT and ALTDLL IP cores and their connections from the rest of the design at the top level.
- Connect extra ports to and from the Altera PHYLite for Parallel Interfaces IP core in the top level design.

Parameter Commonalities and Differences

To ease the manual migration process from the ALTDQ_DQS2 IP core to the Altera PHYLite for Parallel Interfaces IP core, understanding the parameter commonalities and differences of the IP cores is crucial.

These figures and table show the ALTDQ_DQS2 IP core parameters.

Note: For the Altera PHYLite for Parallel Interfaces IP core parameters, refer to **Parameter Settings** on page 32

Figure 20: ALTDQ_DQS2 IP Core Parameter for Arria V and Cyclone V Devices

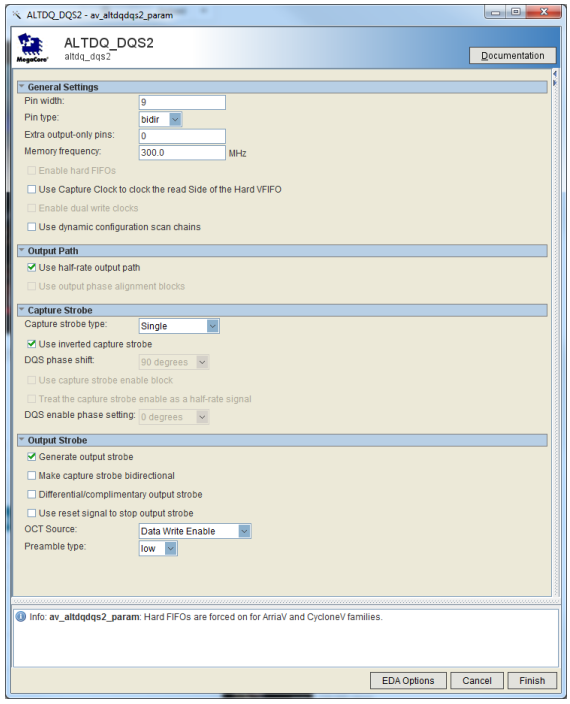
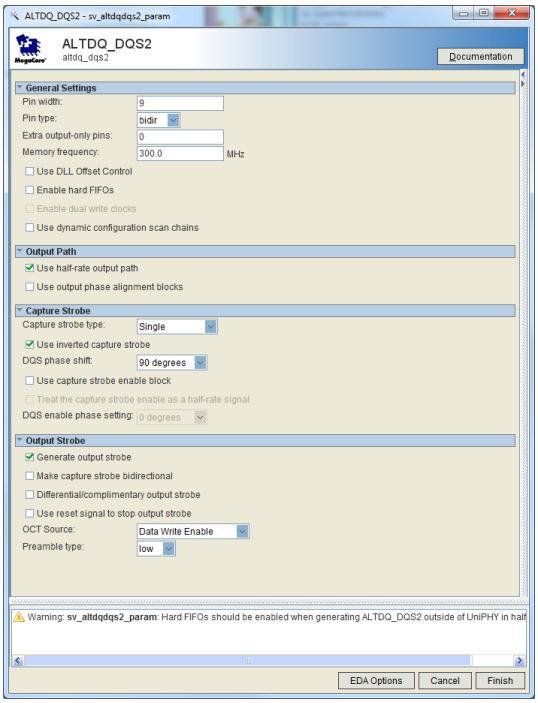


Figure 21: ALTDQ_DQS2 IP Core Parameter for Stratix V Devices



Common Parameters

Table 24: Common Parameters

This table lists the common parameters for the Altera PHYLite for Parallel Interfaces and ALTDQ_DQS2 IP cores.

ALTDQ_DQS2 IP core	Altera PHYLite for Parallel Interfaces IP core
Pin Width	Pin Width
Pin Type	Pin Type
Memory frequency	Memory clock frequency
Use dynamic configuration scan chains	Use dynamic reconfiguration
Use half-rate output path. **Note:** Supports only full or half-rate.	Clock rate of user logic **Note:** Supports only full, half, or quarter-rate.
Capture strobe type **Note:** Supports single, complementary, and differential strobes.	Strobe configuration **Note:** Supports single ended, complementary, and differential strobes.
DQS phase shift	Capture strobe phase shift
Generate Output strobe	Use Output strobe
Differential/Complementary output strobe **Note:** Supports single, complementary, and differential strobes.	Strobe configuration **Note:** Supports single ended, complementary, and differential strobes.
Make capture strobe bidirectional. **Note:** In the Altera PHYLite for Parallel Interfaces IP core, the strobe is bidirectional when the data is bidirectional.	Pin type **Note:** The Use separate strobes parameter can be used for separate input and output strobes with bidirectional data.

Additional Parameter for the Altera PHYLite for Parallel Interfaces IP core

The following figures and table show the additional parameters of the Altera PHYLite for Parallel Interfaces IP core as a result of IP enhancement.

Figure 22: Additional Parameter in General Tab

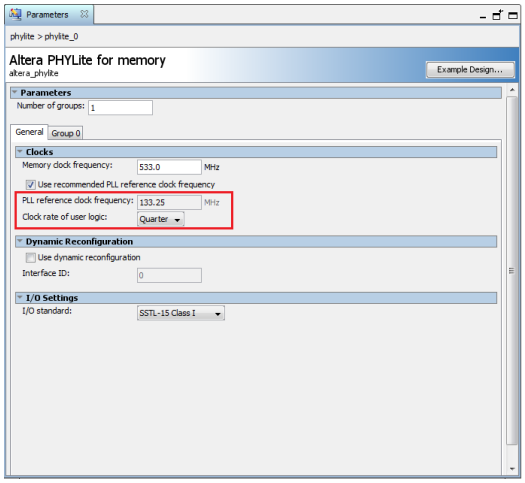


Figure 23: Additional Parameter in Group Tab

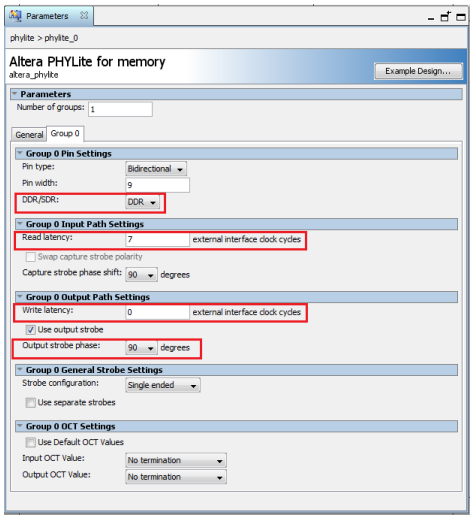


Table 25: Additional Parameters in the Altera PHYLite for Parallel Interfaces IP Core

Parameter	Description
PLL reference clock frequency	Because PLL is available in each I/O bank, you only need to specify the reference clock frequency for the PLL. No PLL instantiation is required.
Clock rate of user logic	The setting (full/half/quarter) is applicable for all groups in the IP core. In the ALTDQ_DQS2 IP core, only full and half rate are supported.
DDR/SDR	This setting is applicable to the respective group only. In the ALTDQ_DQS2 IP core, the default setting is DDR. For Altera PHYLite for Parallel Interfaces IP core, if you need to implement SDR, then in the **DDR/SDR** parameter, select **SDR**.

Parameter	Description
Read latency	The latency between a read command sent to the external device and the first read data returned to the FPGA. This feature internally controls the strobe enable gating.
Write Latency	This is the latency between the write command and the first written data.
Output Strobe Phase	Enables you to set the phase shift between the output strobe and output data. In the ALTDQ_DQS2 IP core, you must ensure phase shifts by generating two clocks with different phases, or manipulating some dynamic reconfiguration settings.

Parameters for ALTDQ_DQS2 IP Core Only

The following figures and table show the parameters supported in the ALTDQ_DQS2 IP core but not in Altera PHYLite for Parallel Interfaces IP core in the Quartus II software version 14.0a10:

Figure 24: ALTDQ_DQS2 IP Core Specific Parameters for Stratix V Devices

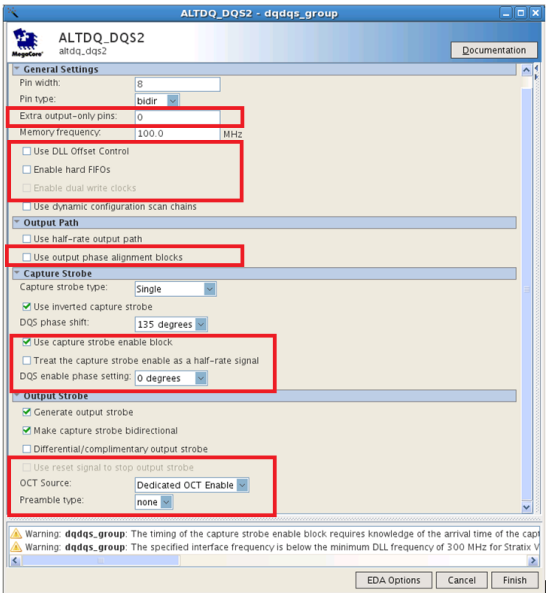


Figure 25: ALTDQ_DQS2 IP Core Specific Parameters for Arria V Devices

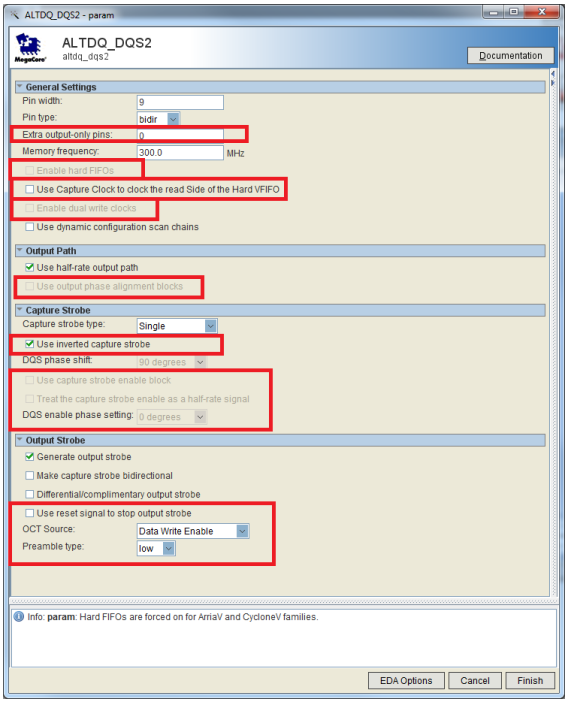


Table 26: ALTDQ_DQS2 IP Core Specific Parameters

Section	Parameter	Description
General Settings	Extra output-only pins	This option is commonly used as datamask pins in the ALTDQ_DQS2 IP core. In the Altera PHYLite for Parallel Interfaces IP core, you may implement this as part of the data group. **Note:** An I/O bank comprises of 48 I/O pins (up to 48 data I/Os as well as the strobe capture logic).
	Use DLL Offset Control	You cannot access the DLL because the DLL is an internal block in Arria 10 devices. **Note:** This feature is only for testing purposes in the ALTDQ_DQS2 IP core.
	Enable hard FIFOs	In Arria 10 devices, the FIFOs are built-in and is always enabled. But you can decide on the full/half/quarter rate conversion to achieve the equivalent data width and rate with the ALTDQ_DQS2 IP core.
	Enable dual write clocks	Enabled the use of separate output clocks for data and strobe in the ALTDQ_DQS2 IP core. Equivalent implementation is available in the Altera PHYLite for Parallel Interfaces IP core. You can control the phase shift with the **Output Strobe Phase** parameter in the Altera PHYLite for Parallel Interfaces IP core.
	Use capture clock to clock the read side of the Hard FIFO (For Arria V and Cyclone V devices only)	This feature is not supported in the Altera PHYLite for Parallel Interfaces IP core. This parameter is only available for Arria V and Cyclone V devices.
Output Path	Use output phase alignment blocks	Arria 10 devices have different architecture.

Section	Parameter	Description
Capture Strobe	Use capture strobe enable block	You cannot access the capture strobe in the Altera PHYLite for Parallel Interfaces IP core because the DQS enable control is an internal block controlled by the read/write latency settings. In Arria 10 devices, the capture strobe enable block is used for input/bidirectional applications. It is controlled by the group's `rdata_en` input port and read latency settings.
	Treat the capture strobe enable as a half-rate signal	
	DQS enable phase setting	
	Use inverted capture strobe	This feature is necessary for QDR protocols. This parameter is only available for Arria V and Cyclone V devices. The **Swap capture strobe polarity** parameter is available for complementary strobe configurations.
Output Strobe	Use reset signal to stop output strobe	There is a reset pin generated in ALTERA_PHYLite instantiation, except that it is for the entire ALTERA_PHYLite reset, rather than allowing stopping of the uni-directional output strobe using this parameter in the ALTDQ_DQS2 IP core.
	OCT source	The OCTs in Arria 10 devices are built-in. You do not need to manually instantiate the OCTs.
	Preamble type	This option is not necessary in the Altera PHYLite for Parallel Interfaces IP core because you can control and customize the output strobe pattern from the core logic using `group_x_strobe_out_in` signal.

Example Design: Manual IP Migration from ALTDQ_DQS2 IP Core to the Altera PHYLite for Parallel Interfaces IP Core

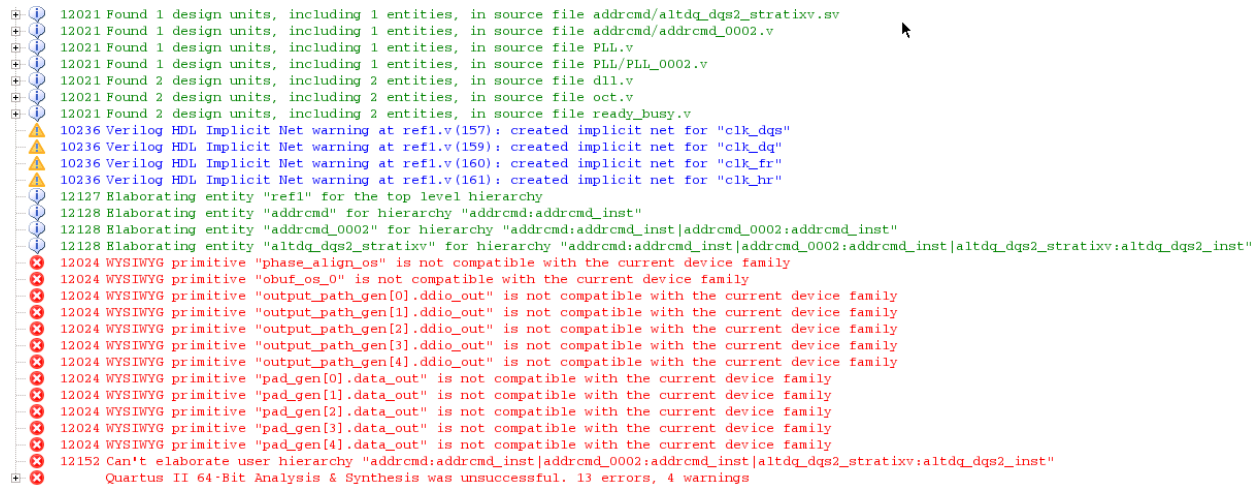
This example design demonstrates how to migrate from the ALTDQ_DQS2 IP core to the Altera PHYLite for Parallel Interfaces IP core on a simple PHY-only design targeting external NAND Flash device. This example design targets a Stratix V device. In Arria V, Cyclone V, and Stratix V devices, a PHY system design that uses the ALTDQ_DQS2 IP core requires other IP cores such as the ALTERA_PLL, ALTDLL, and ALTOCT IP cores.

Note: A proper NAND Flash external memory interface does not require any OCT. However, for general migration illustration purposes, ALTOCT is instantiated in this example design.

To begin, follow these steps:

1. In the Quartus II software version 14.0a10, open the **nand_flash_example_14.0a10.qar**.
2. In the Quartus II dialog box, click **Yes**.
3. Run **Analysis and Synthesis**. The following figure shows the generated error messages, indicating that there is no direct migration support from the ALTDQ_DQS2 IP core to the Altera PHYLite for Parallel Interfaces IP core. To manually migrate to the Altera PHYLite for Parallel Interfaces IP core, understanding the ALTDQ_DQS2 design is crucial.

Figure 26: Error Message


12021 Found 1 design units, including 1 entities, in source file addrcmd/altdq_dqs2_stratixv.sv
12021 Found 1 design units, including 1 entities, in source file addrcmd/addrcmd_0002.v
12021 Found 1 design units, including 1 entities, in source file PLL.v
12021 Found 1 design units, including 1 entities, in source file PLL/PLL_0002.v
12021 Found 2 design units, including 2 entities, in source file dll.v
12021 Found 2 design units, including 2 entities, in source file oct.v
12021 Found 2 design units, including 2 entities, in source file ready_busy.v
10236 Verilog HDL Implicit Net warning at ref1.v(157): created implicit net for "clk_dqs"
10236 Verilog HDL Implicit Net warning at ref1.v(159): created implicit net for "clk_dq"
10236 Verilog HDL Implicit Net warning at ref1.v(160): created implicit net for "clk_fr"
10236 Verilog HDL Implicit Net warning at ref1.v(161): created implicit net for "clk_hr"
12127 Elaborating entity "ref1" for the top level hierarchy
12128 Elaborating entity "addrcmd" for hierarchy "addrcmd:addrcmd_inst"
12128 Elaborating entity "addrcmd_0002" for hierarchy "addrcmd:addrcmd_inst|addrcmd_0002:addrcmd_inst"
12128 Elaborating entity "altdq_dqs2_stratixv" for hierarchy "addrcmd:addrcmd_inst|addrcmd_0002:addrcmd_inst|altdq_dqs2_stratixv:altdq_dqs2_inst"
12024 WYSIWYG primitive "phase_align_os" is not compatible with the current device family
12024 WYSIWYG primitive "obuf_os_0" is not compatible with the current device family
12024 WYSIWYG primitive "output_path_gen[0].ddio_out" is not compatible with the current device family
12024 WYSIWYG primitive "output_path_gen[1].ddio_out" is not compatible with the current device family
12024 WYSIWYG primitive "output_path_gen[2].ddio_out" is not compatible with the current device family
12024 WYSIWYG primitive "output_path_gen[3].ddio_out" is not compatible with the current device family
12024 WYSIWYG primitive "output_path_gen[4].ddio_out" is not compatible with the current device family
12024 WYSIWYG primitive "pad_gen[0].data_out" is not compatible with the current device family
12024 WYSIWYG primitive "pad_gen[1].data_out" is not compatible with the current device family
12024 WYSIWYG primitive "pad_gen[2].data_out" is not compatible with the current device family
12024 WYSIWYG primitive "pad_gen[3].data_out" is not compatible with the current device family
12024 WYSIWYG primitive "pad_gen[4].data_out" is not compatible with the current device family
12152 Can't elaborate user hierarchy "addrcmd:addrcmd_inst|addrcmd_0002:addrcmd_inst|altdq_dqs2_stratixv:altdq_dqs2_inst"
Quartus II 64-Bit Analysis & Synthesis was unsuccessful. 13 errors, 4 warnings

Basic NAND Flash Protocol

Table 27: Details of the Micron MT29F NAND Flash Memory Datasheet

This table lists the details of the Micron MT29F NAND Flash memory datasheet (focusing on synchronous mode):

Synchronous Signals	Description
Memory frequency	100MHz
Data pin width	8-bit
Data Mode	DDR
DQx and DQS	<table><tr><th>Signal</th><th>Type</th><th>Description</th></tr><tr><td>DQx</td><td>I/O</td><td>**Data inputs/outputs**. The bidirectional I/Os transfer address, data, and command information.</td></tr><tr><td>DQS</td><td>I/O</td><td>**Data strobe**. Provides a synchronous reference for data input and output.</td></tr></table>

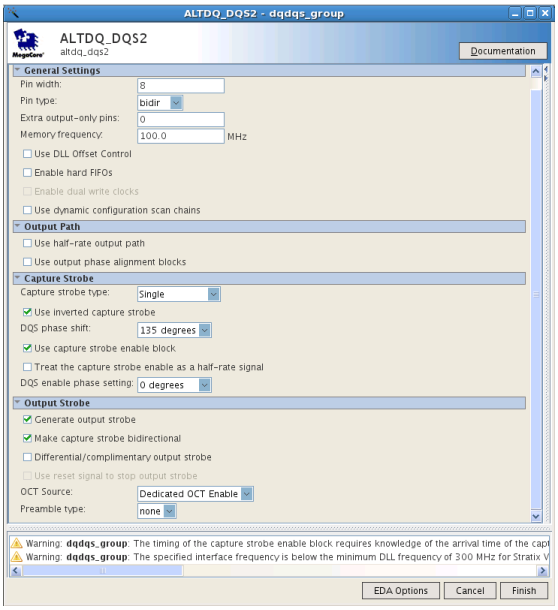
<table>
<tr><th>Synchronous Signals</th><th>Description</th></tr>
<tr><td>CMD/ADDR signals (output from FPGA, input to memory)</td><td>
<table>
<tr><th>Signal</th><th>Type</th><th>Description</th></tr>
<tr><td>ALE</td><td>Input</td><td>Address latch enable. Loads an address from DQx into the address register.</td></tr>
<tr><td>CE#</td><td>Input</td><td>Chip enable. Enables or disables one or more die (LUNs) in a target.</td></tr>
<tr><td>CLE</td><td>Input</td><td>Command latch enable. Loads a command from DQx into the command register.</td></tr>
<tr><td>W/R#</td><td>Input</td><td>Read enable and write/read enable. RE# transfers serial data from the NAND Flash to the host system when the asynchronous interface is active. When the synchronous interface is active, WR# controls the direction of DQx and DQS.</td></tr>
<tr><td>CLK</td><td>Input</td><td>Write enable and clock. WE# transfers commands, addresses, and serial data from the host system to the NAND Flash when the asynchronous interface is active. When the synchronous interface is active, CLK latches command and address cycles.</td></tr>
<tr><td>WP#</td><td>Input</td><td>Write protect. Enables and disables array PROGRAM and ERASE operations.</td></tr>
</table>
</td></tr>
<tr><td>R/B# (output from memory, input to the FPGA.)</td><td>Ready/Busy. This signal is an open-drain, active-low output that requires an external pull-up resistor. This signal indicates target array activity.</td></tr>
</table>

Implementation Using the ALTDQ_DQS2 IP Core

The following lists the possible implementations when you target Arria V, Cyclone V, and Stratix V devices:

- Instantiates two ALTDQ_DQS2 IP cores.
  - Bidirectional type for DQ and DQS (**Figure 27**)
  - Output type for Addr/Cmd (**Figure 28**)
- Instantiates a ALTIOBUF (input) for the `ready` signal
- Connects the ALTERA_PLL, ALTDLL, and ALTOCT IP cores to complete the PHY-only design.

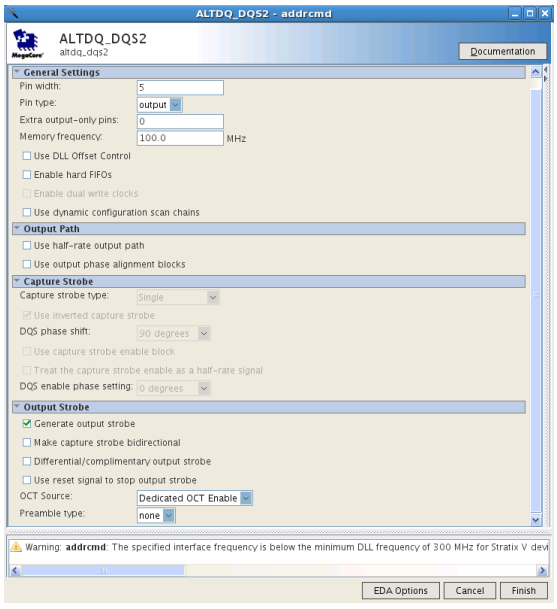
Figure 27: ALTDQ_DQS2 Settings for Bidirectional Type DQ and DQS


ALTDQ_DQS2 - dqdqs_group
ALTDQ_DQS2
altdq_dqs2
Documentation
General Settings
Pin width: 8
Pin type: bidir
Extra output-only pins: 0
Memory frequency: 100.0 MHz
Use DLL Offset Control
Enable hard FIFOs
Enable dual write clocks
Use dynamic configuration scan chains
Output Path
Use half-rate output path
Use output phase alignment blocks
Capture Strobe
Capture strobe type: Single
Use inverted capture strobe
DQS phase shift: 135 degrees
Use capture strobe enable block
Treat the capture strobe enable as a half-rate signal
DQS enable phase setting: 0 degrees
Output Strobe
Generate output strobe
Make capture strobe bidirectional
Differential/complimentary output strobe
Use reset signal to stop output strobe
OCT Source: Dedicated OCT Enable
Preamble type: none
Warning: dqdqs_group: The timing of the capture strobe enable block requires knowledge of the arrival time of the capt
Warning: dqdqs_group: The specified interface frequency is below the minimum DLL frequency of 300 MHz for Stratix V
EDA Options
Cancel
Finish

Note: The DQS enable block must be enabled for NAND Flash, which has bidirectional strobe.

Note: DQS phase shift is set to 135° to gain maximum margin (due to the memory clock is slower than the DLL's minimum frequency)

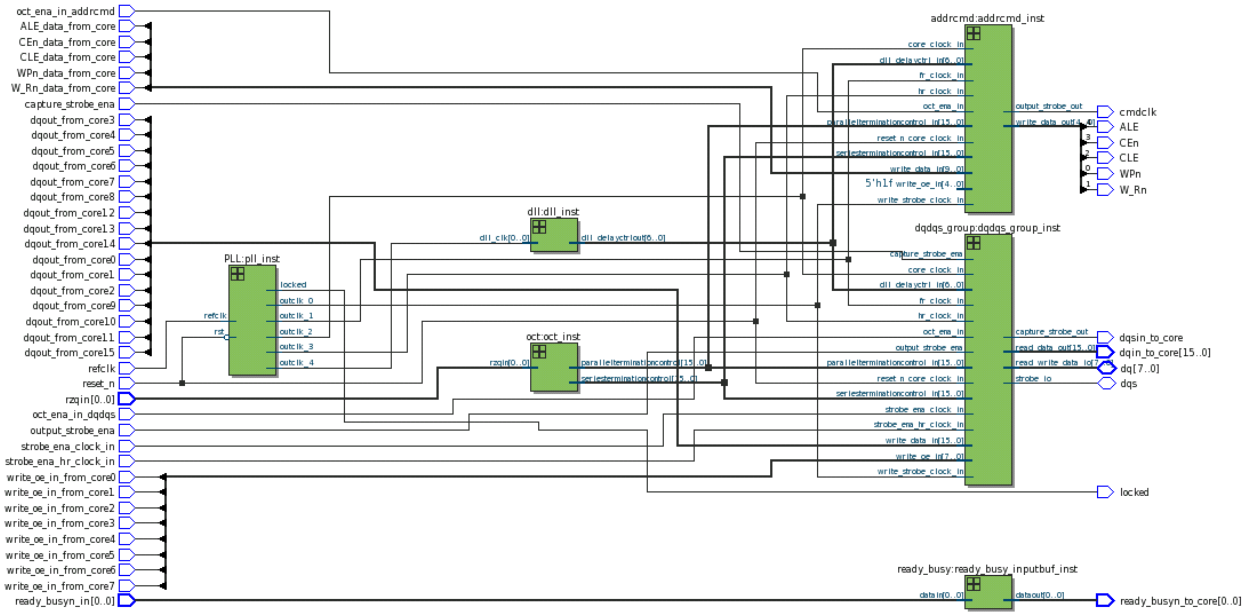
Figure 28: ALTDQ_DQS2 settings for output type (Addr/Cmd)



Note: The settings in the figure are for the address/command lines.

The following figure shows the RTL viewer for a NAND Flash simple design based on the ALTDQ_DQS2 IP core from this implementation.

Figure 29: RTL viewer for a NAND Flash simple design based on ALTDQ_DQS2



Implementation using the Altera PHYLite for Parallel Interfaces IP Core

You can configure the IP in a single Altera PHYLite for Parallel Interfaces IP core for multiple groups (maximum 48 I/O pins each), instead of instantiating multiple ALTDQ_DQS2 configurations for various settings/group.

The following lists the possible implementations when you target Arria 10 devices:

- Instantiates one Altera PHYLite for Parallel Interfaces IP core with three groups
  - Bidirectional type for DQ and DQS
  - Output type for Addr/Cmd
  - Input type for the Ready signal

Note: Each group in the Altera PHYLite for Parallel Interfaces IP core can have 48 I/Os, and the IP supports up to 18 groups.

Figure 30: General Tab Settings

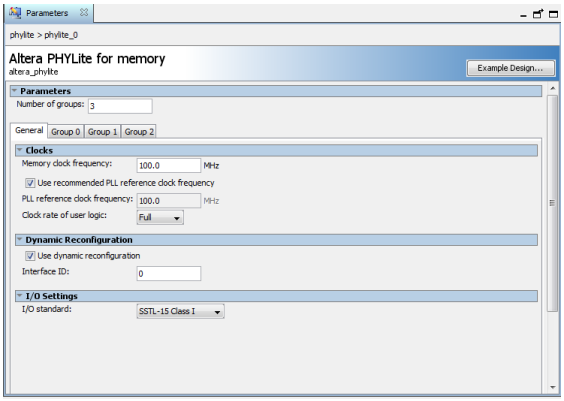


Figure 31: Group 0 settings (Bidirectional type for DQ and DQS)

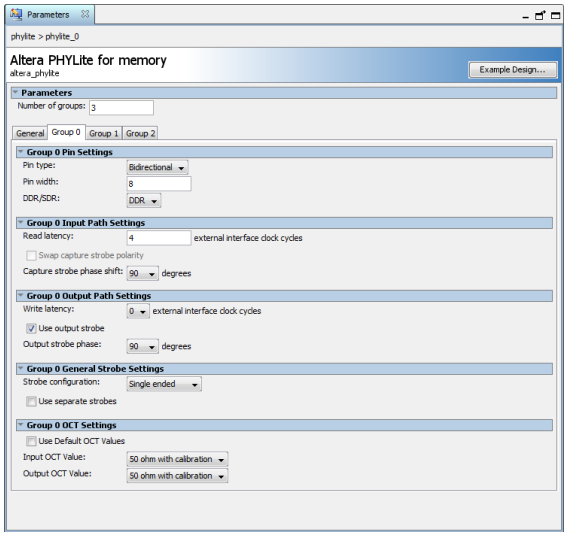


Figure 32: Group 1 settings (Output type for Addr/Cmd)

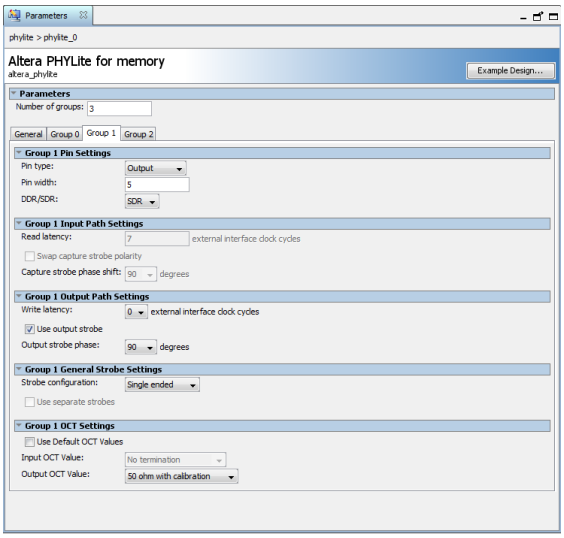
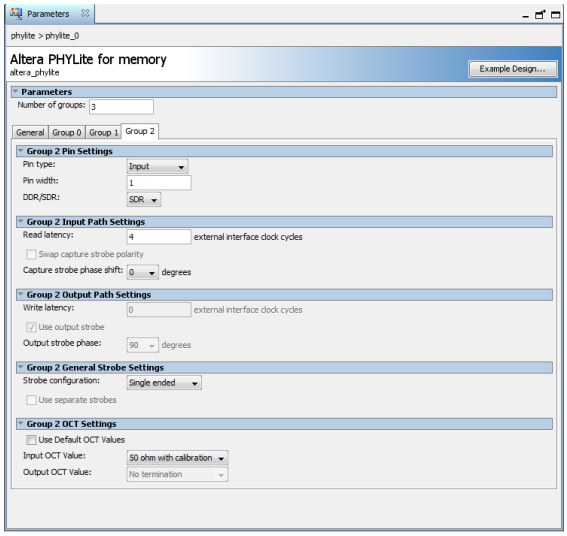
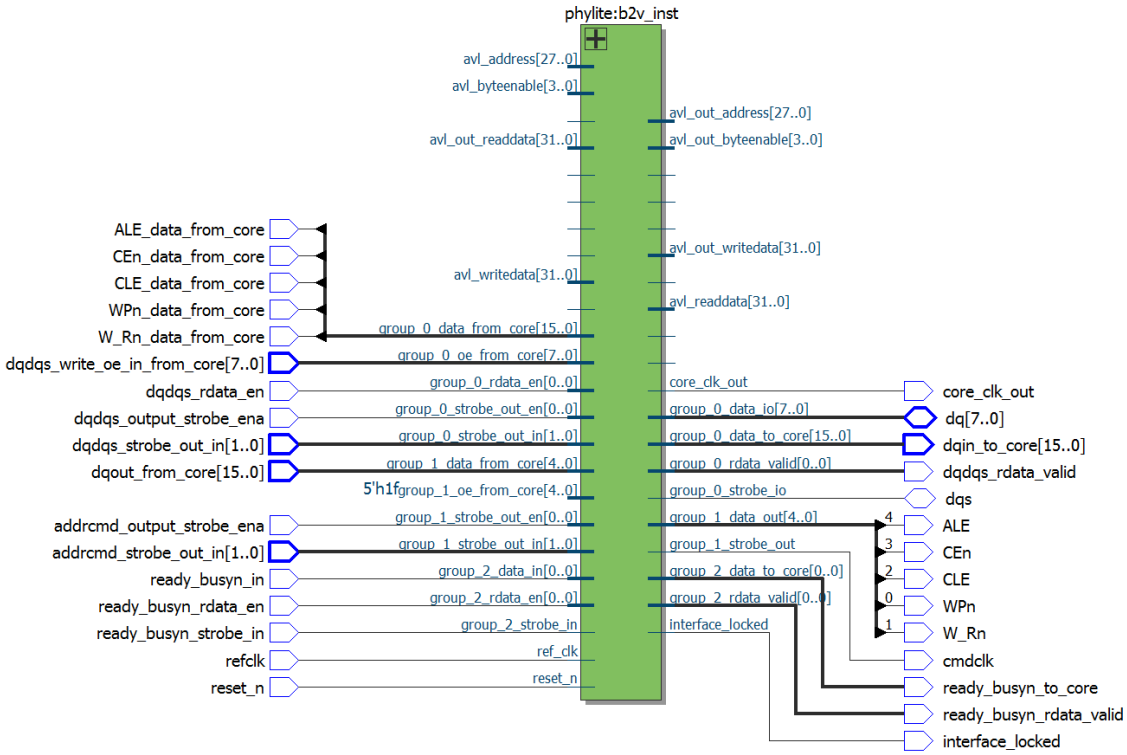


Figure 33: Group 2 settings (Input type for the Ready signal)



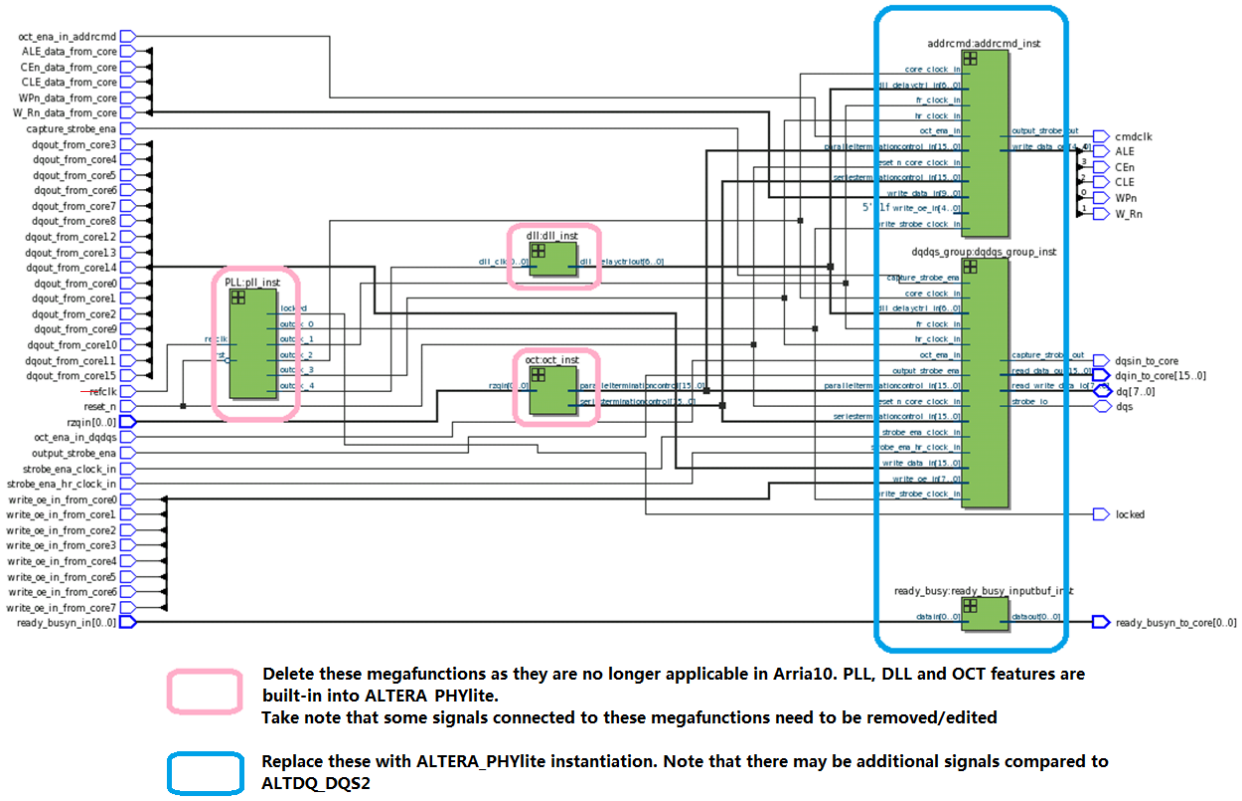
The following figure shows the RTL viewer for a NAND Flash simple design based on the Altera PHYLite for Parallel Interfaces IP core implementation above.

Figure 34: RTL Viewer for a NAND Flash Simple Design Based on the Altera PHYLite for Parallel Interfaces IP Core



Manual Migration between ALTDQ_DQS2 and Altera PHYLite for Parallel Interfaces IP Cores

Figure 35: Migration Process Overview for the NAND Flash Simple Design



1. After generating and instantiating the equivalent Altera PHYLite for Parallel Interfaces IP core, delete the ALTERA_PLL, ALTDLL and ALTOCT IP cores.
2. Remove the connections between the ALTERA_PLL, ALTDLL, ALTOCT IP cores and the rest of the design in the RTL.

Table 28: List of Signals for Connection Removal

Signal	Description
`oct_ena_in`	Remove this connection. The OCT is built-in in Altera PHYLite for Parallel Interfaces IP core.
`strobe_ena_clock_in`	Remove this connection. The DQS enable is built-in in Altera PHYLite for Parallel Interfaces IP core.
`strobe_ena_hr_clock_in`	Remove this connection. The DQS enable is built-in in Altera PHYLite for Parallel Interfaces IP core.
`capture_strobe_out`	Remove this connection. This signal is not available for Arria 10 devices.

3. Connect the Altera PHYLite for Parallel Interfaces IP core signals appropriately to the design in the RTL. The `core_clock` and `rdata_valid` signals are examples of additional signal from the

ALTERA_PHYlite which will be feeding the core logic. The following table lists information about connecting similar or new signals.

Table 29: Connecting Similar or New Signals

Signal	Description
rdata_en	In the Altera PHYLite for Parallel Interfaces IP core, this signal is similar but not exactly the same as the capture_strobe_ena in the ALTDQ_DQS2 IP core. Connect this signal to the core. This signal must be held high for the number of expected read words after a read command.
rzqin	You must manually create this signal in the Altera PHYLite for Parallel Interfaces IP core using the QSF assignments in Arria 10 devices. Refer to QSF Assignments and Design Example sections in Altera OCT Megafunction User Guide for more information on how to create rzqin signal.
core_clk_out	A new signal for the Altera PHYLite for Parallel Interfaces IP core. This signal was created because the PLL is built-in for Arria 10 devices.
rdata_valid	This signal is a delayed rdata_en signal by READ_LATENCY + 6. Because the data_to_core signal is valid on rdata_valid, feed the rdata_valid signal to the core appropriately to know when the correct data_to_core is expected. Always matches the rdata_en alignment. Refer to **Input Path** on page 8 section.
strobe_out_in	Use this signal to send customized strobe data pattern. This signal is synchronous to the core_clk output from the IP core. This signal is controlled from the core.
strobe_out_en	This signal enables the output strobe. This signal is controlled from the core.

4. Run **Analysis and Elaboration** to confirm that the manual IP migration is successful.

Related Information

Altera OCT Megafunction User Guide

Document Revision History

Table 30: Document Revision History

Date	Version	Changes
June 2015	2015.06.12	• Updated Avalon Address R/W from 3'h2 to 3'h4 for all features in Address Map table. • Added new parameter **Use core PLL reference clock connection** and **Data configuration** in Altera PHYLite for Parallel Interfaces IP Core Parameter Settings table. • Updated values in VCO Frequency Multiplication Factor table.
January 2015	2015.01.28	Updated related information link to Functional Description for External Memory Interfaces in Arria 10 Devices.
December, 2014	2014.12.30	• Updated the name of the IP core from Altera PHYLite for Memory to Altera PHYLite for Parallel Interfaces. • Updated the maximum clock frequency from 800 MHz to 1333.333 MHz. • Clarified that to achieve timing closure at 800 MHz and above, you must use dynamic reconfiguration to calibrate the interface. • Added `data_out_n`/`data_io_n` signals to the Output Path Signals table. • Added `data_in_n`/`data_io_n` signals to the Input Path Signals table. • Updated `data_out`/`data_io` and `data_in`/`data_io` signals in the Input Path Signals and Output Path Signals tables. • Updated Parameter Settings table to include **Group <x> Timing Settings** information. • Updated Timing section to include **Input Strobe Setup Delay Constrain** and **Input Strobe Hold Delay Constrain** parameters information.
August, 2014	2014.08.18	• Renamed the term megafunction to IP core. • Added information about output path data alignment, input path data alignment, OCT, I/O standards, placement restrictions, timing, dynamic reconfiguration. • Added the **PHYLite_delay_calculations.xlsx** file. • Replaced **ALTERA_PHYLite_nand_flash_example_131a10.qar** file with **nand_flash_example_14.0a10.qar** file.
November, 2013	2013.11.29	Initial release.